DESIGN AND ANALYSIS OF DOUBLE-LAYER GRIDS

By

WANG-REI TANG
Bachelor of Science in Architecture and Architectural Engineering
Technical University of Budapest
June 1998

Submitted to the Department of Civil and Environmental Engineering
In Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING
IN CIVIL AND ENVIRONMENTAL ENGINEERING

At the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 1999

Signature of the Author_______________________________________
Department of Civil and Environmental Engineering
May 7, 1999

Certified by_______________________________________
Professor Jerome Connor
Department of Civil and Environmental Engineering
Thesis Supervisor

Accepted by_______________________________________
Professor Andrew J. Whittle
Chairman, Department Committee on Graduate Studies

DESIGN AND ANALYSIS OF DOUBLE-LAYER GRIDS

By

WANG-REI TANG

Submitted to the Department of Civil and Environmental Engineering on May 7, 1999 in partial fulfillment of the requirements for the degree of Master of Engineering in Civil and Environmental Engineering

Abstract

This thesis focuses on the design and analysis of double-layer structural grids. The analysis is based on the use of formex algebra. Basically, formex algebra is a mathematical system designed to deal in a convenient way with problems of data preparation and graphics in computer aided design processes. Formex algebra is particularly useful for space structures having complex geometry. Following the formex formulation, one case study is considered. Two models of double-layer grids are designed and analyzed using SAP and ADINA.

Thesis Supervisor: Jerome Connor
Title: Professor of Civil and Environmental Engineering

Acknowledgements

First of all, I would like to thank my advisor, Professor Jerome Connor, for all his help during the entire year, for his understanding and trust in whatever I was doing. Thanks to Charlie Helliwell for his support to the M.Eng. program in general, and to the HPS project and thesis in particular. Thanks to the structure group for their contribution to our group project. Thanks to all my classmates in the M.Eng group for all the happiness we spend together.

Very special thanks to my parents, brother and for their continuous support for whatever decision I made. Thanks to my relatives and friends for their support through all these years.

Finally, I thank God for all the wonderful things that happened to me.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER ONE BACKGROUND

In the past, space structures were regarded as exotic and unconventional. Now, they are used frequently in many countries around the world, not only because they are aesthetically pleasing but also because they are efficient structural forms.

Over the last century, engineers have been striving to build larger and larger structures with less material and less cost. Designers agree that for large spans, conventional beam and truss structures prove to be uneconomical, and have turned to three-dimensional structures which have incredible rigidity and the ability to cover large spans with minimum weight.

Double-layer grids have developed rapidly in the past several decades. They are suited for covering exhibition pavilions, swimming pools, and many types of industrial buildings in which large unobstructed areas are required. Experience shows that in many countries double-layer grid structures can compete very successfully with more conventional systems. Due to the complex and regular nature of double layer structural grids, it is convenience to use computer techniques for the design and production process. One of the main problems in design is generating the data required for the phases of the numerical model used in analysis models, and in general, dealing with large amounts of data. The data required by a numerical model of a reasonably large structure is substantial. The input includes the nodal coordinates, the connectivity, element properties, boundary or support conditions and the applied loads. Formex algebra may be used in various ways to overcome the difficulties of data preparation. In particular, the description of the interconnection pattern of a structural system may be conveniently formulated through the concepts of formex algebra.

CHAPTER TWO FUNDAMENTALS OF FORMEX ALGEBRA

2.1 Introduction

Formex algebra is a mathematical system that consists of a set of definitions for numerical objects, and a set of rules for manipulating these objects. Formex algebra is the basis of this paper and all the techniques that simplify the work of data preparation and graphics in computer aided design processes. Formex algebra allows all kinds of networks to be formulated conveniently. In what follows, the basic definitions and rules are introduced.

2.2 Signets

A 'signet' is an object of the form

$$[I_1, I_2, I_3, I_4, \ldots\ldots, I_n]$$

where n is greater or equal to 1 and where $I_1, I_2, I_3, I_4, \ldots\ldots, I_n$ are integers. For example,

$$[3]$$

$$[5, 2]$$

$$[-4, -3, 9]$$

and

$$[45, 98, -3, 8, -45]$$

are signets.

The integers that a signet contains are called 'indices' and the number of indices in a signet is called the 'grade' of the signet. For example, [4, 7, 11, 1] is said to be a signet.

‘4’, ‘7’, ‘11’ and ‘1’ are said to be the indices of the signet. Since there are four integers in the signet, the signet is of the grade 4.

In order to have a relationship between two signets, such as ‘equal’, ‘greater than’ or ‘less than’, the two signets must be in the same grade. Two signets of the same grade are said to be ‘equal’ if and only if they are identical. For example,

$$[I_1, I_2, I_3, I_4] = [4, 7, 11, 1]$$

shows that $I_1 = 4$, $I_2 = 7$, $I_3 = 11$, and $I_4 = 1$.

A signet, $s' = [I_1, I_2, I_3, I_4, \ldots, I_n]$, is said to be greater than another signet $s = [I'_1, I'_2, I'_3, I'_4, \ldots, I'_n]$, if and only if, for a value of $i = 1, 2, 3, \ldots, n$

$$I_i > I'_i$$

And at the same time, for all integer values of j, where $1 \leq j < i$

$$I_j = I'_j$$

For example, let s = [4, 7, 11, 1] and s’ = [4, 7, 12, 2]. The s’ is greater than s since the third index of s’ is greater than the third index of s and the first and the second indices of both signets are equal. Other examples are

$$[4, -3, 4, -88] > [3, 7, 1, -5]$$

$$[7, 8, 34] > [7, 7, 56]$$

$$[3, 6] > [3, 3]$$

and

$$[0, 0, 5, 5] > [0, 0, 5, 4]$$

Furthermore, if

$$[\beta, 2, -8] \geq [9, 2, -8]$$

then $\beta \geq 9$.

If s and s' are two signets of the same grade, and the indices of both signets are unsigned single digits, it is important to note that we can find out the relationship between two signets by taking out the inter-indexical commas. For instance,

$$[1, 9, 9, 9]$$

is greater than

$$[1, 9, 9, 8]$$

just as 1999 is greater than 1998

Also,

$$[3, 4, 5]$$

is less than

$$[3, 2, 6]$$

is just as 345 is less than 326.

2.3 Cantles

A cantle is an object of the form

$$[s_{1;}\ s_{2;}\ s_{3;}\ s_{4;\ldots\ldots;}\ s_m]$$

where $m \geq 1$ and where $s_{1,}\ s_{2,}\ s_{3,}\ s_{4,\ldots\ldots,}\ s_m$ are signets of the same grade. The number of signets in a cantle is called the 'plexitude' of the cantle. The common grade of the signets within a cantle is also the 'grade' of the cantle. When a cantle is written down explicitly, the enclosing square brackets of its signets can be ignored. A cantle of the m-th plexitude is called as a m-plex cantle. Thus,

$$[4, 2, 5; 5, 5, 3; -8, 0, 88]$$

is a cantle of the third plexitude and third grade,

$$[2, -5; 4, -99; 33, 34]$$

is a cantle of the third plexitude and second grade,

$$[6; 7; -3; 0]$$

is a cantle of the fourth plexitude and first grade,

$$[25, 67, 7, 8; 32, 6, 6, 35]$$

is a cantle of the second plexitude and fourth grade,

$$[4]$$

is a signet and also a cantle,

$$[4, 3]$$

is a cantle of the first plexitude and second grade. A cantle consists of a multiple of signets. It can consist of a single signet. The word 'signet' is also called the cantle of the first plexitude.

Two cantles are said to be 'equal' if and only if they are identical. For example,

$$[s_1; s_2; s_3; s_4] = [4, 2, 5; 5, 5, 3; -8, 0, 88; -4,3, 2]$$

shows that $s_1 = [4, 2, 5]$, $s_2 = [5, 5, 3]$, $s_3 = [-8, 0, 88]$, and $s_4 = [-4, 3, 2]$.

A cantle $c = [s_1; s_2; s_3; s_4; \ldots; s_m]$ is called regular, if and only if,

$$s_1 \leq s_2 \leq s_3 \leq s_4 \leq \ldots \leq s_m$$

A cantle is said to be irregular otherwise. A cantle of the first plexitude is considered to be regular.

A cantle c is said to be the variant of another cantle c' if and only if they are of the same plexitude and the same grade, and c is equal to c' after a rearrangement of the signets of c. For example,

$$C1 = [3, 4, 5; 5, 6, 7; 2, 3, 4; 4, 4, 4]$$

$$C2 = [3, 4, 5; 4, 4, 4; 5, 6, 7; 2, 3, 4]$$

C1 is said to be the variant of C2, and C2 is said to be the variant of C1 as well. Any cantle is always considered to be the variant of itself.

2.4 Formices

A 'formex' is an object of the form

$$\{c_1, c_2, c_3, c_4, \ldots, c_r\}$$

where $r \geq 1$ and $c_1, c_2, c_3, c_4, \ldots, c_r$ are cantles of the same grade. The number of cantles in a formex is called the 'order' of the cantle. And the serial position number of a cantle in a formex is called the 'orderate' of that cantle. The common grade of the cantles within a formex is also the 'grade' of the formex. A formex must contain the cantles with the same grade, but the plexitude of each cantle doesn't have to be the same. For example,

$$\{[5, 6; 7, 9], [2, 9; 0, -4; 7, 3; -45, -98], [78, 5], [38,49]\}$$

is a formex of the fourth order and the second grade, and the orderate of [38,49] and [2, 9; 0, -4; 7, 3; -45, -98] in

$$\{[5, 6; 7, 9], [2, 9; 0, -4; 7, 3; -45, -98], [78, 5], [38,49]\}$$

are 4 and 2. Another example,

$$\{[5, 6,0; 7, 9,77], [2, 9, 0; -4, 7, 3], [-45, -98, 7]\}$$

is a formex of the third order and the third grade, and the orderate of [5, 6,0; 7, 9,77] in

$$\{[5, 6,0; 7, 9,77], [2, 9, 0; -4, 7, 3], [-45, -98, 7]\}$$

is 1.

If all cantles in a formex are regular, then this formex is said to be regular too. Otherwise, it is said to be irregular. For instance, {[5, 6; 7, 9], [2, 9; 3, 0; 3, 3; 4, 5], [78, 79], [38,49]}

is said to be regular, and

{[5, 6; 7, 9], [2, 9; 0, -4; 7, 3; -45, -98], [78, 5], [38,49]}

is said to be irregular.

Let F = {c_1, c_2, c_3, $c_{4,\ldots\ldots,}$ c_r} and F' = {c'_1, c'_2, c'_3, $c'_{4,\ldots\ldots,}$ c'_r}. If the two formices are of the same order and grade, the formex F is said to be the 'variant' of F' if and only if c_i is the variant of c'_i for i = 1, 2, 3, 4,.....,r. For example, consider F = {[1, 3; 4, 4], [3, 2], [77, 45; 34, 55; 245, 546; 6, 0]}

and

F1 = {[4, 4; 1, 3], [3, 2], [245, 546; 34, 55; 77, 45; 6, 0]}

the formex F1 is said to be the variant of F and F is a variant of F1.

A formex F is said to be a 'sequation' of another F' if and only if they are of the same order and the same grade, and F is equal to F' after a rearrangement of the cantles of F. For instance, consider

F1 = {[1, 3; 4, 4], [3, 2], [77, 45; 34, 55; 245, 546; 6, 0]}

and

F2 = {[1, 3; 4, 4], [77, 45; 34, 55; 245, 546; 6, 0], [3, 2]}

F1 is said to be the sequation of F2, and F2 is said to be the sequation of F1 as well. Any formex is always considered to be the sequation of itself.

If all the cantles in a formex have the same plexitude, then the formex is said to be 'homogeneous'; otherwise it is said to be 'nonhomogeneous'. A homogeneous formex whose cantles are of the mth plexitude is referred to as a homogeneous formex of the mth plexitude. For example, {[2, 3, 4; 3, 4, 5], [1, 2, 3; 2, 2, 2], [5, 5, 6; -5, -9, 0]}

is a homogeneous formex of the second plexitude. The empty formex is considered to be a homogeneous formex of arbitrary plexitude. A homogeneous formex of the mth plexitude may be referred to a s an m-plex formex.

A homogeneous formex of the first plexitude is referred to as an 'ingot'. For example,

{[1, 2, 3], [4, 5, 6], [7, 8, 9]}

is an ingot. Another easier way to see if a formex is ingot or not is to see if ';' exist in a formex or not.

Let A, B, C, D, E, F, G and H be signets of the same grade. Let F be a formex and is in the form of {[A], [A; B], [C; D; E], [E; D]}. An ingot is said to be a 'catena' of F if, and ony if, every signet of F is in this ingot. For example, G is an ingot of the same grade as F, and

G = {[A],[B],[C],[D],[E]},

So G is a catena of F.

If an ingot is a catena of F, and this ingot is said to be an 'exclusive' catena of F if and only if every signet of this ingot is in F, and is said to be an 'inclusive' catena otherwise. So G in this case is an exclusive catena of F. Next example,

E = {[A],[B],[C],[D],[E],[F],[G]}

so E is an inclusive catena of F. More examples,

Q = {[5, 1], [3, 4; 2, 6], [7, 4],[6, 2; 5, 1]}

then

{[5, 1], [3, 4], [2, 6], [7, 4], [6, 2]}

is an exclusive catena of Q and

$$\{[5, 1], [2, 5], [2,6], [7, 4], [3, 4], [6, 2]\}$$

is an inclusive catena of Q.

2.5 Composition of formices

The composition of two formices of the same grade, F1 and F2, is written in the form of

$$F = F1 \# F2$$

F is formed from F1 at first, taking the cantles of F1, and then followed by taking the cantles of F2. For instance, consider

$$F1 = \{[1, 2, 3; 4, 5, 6; 9, 0, 11], [4, 4, 6]\}$$

and

$$F2 = \{[3, 5, 56; 65, 66, 66], [4, 6, 5; 90, 0, 5]\}$$

then

$$F1 \# F2 = \{[1, 2, 3; 4, 5, 6; 9, 0, 11], [4, 4, 6], [3, 5, 56; 65, 66, 66], [4, 6, 5; 90, 0, 5]\}$$

and

$$F2 \# F1 = \{[3, 5, 56; 65, 66, 66], [4, 6, 5; 90, 0, 5], [1, 2, 3; 4, 5, 6; 9, 0, 11], [4, 4, 6]\}$$

The symbol # is referred to as the 'duplus symbol' and is read as 'duplus'. So F1 # F2 is read as 'F1 duplus F2'. During a composition, the term F1 which is on the left side of # should be always taken first and then followed by the term F2 which is on the right side of #. That is why F1 # F2 is not equal F2 # F1 in general. A formex composition is not commutative.

If F1, F2 and F3 are formices of the same grade, then

$$F1 \# (F2 \# F3) = (F1 \# F2) \# F3$$

The formex composition is associative.

If F1 and F2 are formices of the same grade, then the formices (F1 # F2) and (F2 # F1) are sequations of each other.

For any formex F

$$F \# \{ \} = \{ \} \# F = F$$

2.6 Libra notation

If formex F_i = {[i, i+1; i, i-1], [i-1, i+1; i, i], and i is any integer, then

$$F_1 = \{[1, 2; 1, 0], [0, 2; 1, 1]$$

$$F_2 = \{[2, 3; 2, 1], [1, 3; 2, 2]$$

and

$$F_1 \# F_2 = \{[1, 2; 1, 0], [0, 2; 1, 1], [2, 3; 2, 1], [1, 3; 2, 2]\}$$

The other way to show F_1 # F_2 is to use the libra notation.

$$\Sigma\frac{2}{i=1}F_i$$

The symbol Σ— is referred as the 'libra symbol' and

$$\Sigma\frac{n}{i=m}$$

Is read as 'libra i=m to n'. Also, the notation used in writing libra composition is referred to as the 'libra notation'.

F_i is a formex corresponds to *i*, and m and n are any integers which represents the range of i from m to n, for example, taking i = m, m+1, m+2,....., n-2, n-1, n if m is greater than n. More specifically, if m < n then

$$\Sigma\frac{n}{i=m}F_i = F_m \# F_{m+1} \# ..... \# F_{n-1} \# F_n$$

And if m = n then

$$\sum_{i=m}^{n} F_i = F_m$$

And if m > n then

$$\sum_{i=m}^{n} F_i = F_m \# F_{m-1} \# \ldots\ldots \# F_{n+1} \# F_n$$

For example, if F_i ={[i, i+1; i, i-1], [i-1, i+1; i, i]} then

$$\sum_{i=2}^{3} F_i = \{[2,3;2,1],[[1,3;2,2],[3,4;3,2],[2,4,3,3]]$$

and

$$\sum_{i=3}^{2} F_i = \{[3,4;3,2],[2,4;3,3],[2,3;2,1],[1,3;2,2]\}$$

Also,

$$\sum_{i=-1}^{1} [i+i,1] = \{[-2,-1],[0,0],[2,1]\}$$

CHAPTER THREE FORMEX PLOTS

3.1 Introduction

A formex can be graphically represented, and the resulting configuration is called as a formex plot.

The relationship between formices and geometric configurations is very important in the practical applications of formex algebra. A fromex can be plotted in different ways and the resulting configuration may still have some similarity to each other.

3.2 Retrobases

Consider F is a formex of the seventh order and the second grade and is in the form of

F = {[1, 2; 2, 2], [2, 3; 2, 2], [2, 1; 2, 2], [1, 2; 2, 1], [4, 2; 3,2], [2, 2; 3, 2], [3, 2; 2, 3]}

Let x and y be a function of every signet $[S_1, S_2]$, and

$$x = (2S_1-1)/4$$

$$y = S_2/4$$

Let x and y be the center of each circle in a two-dimensional Cartesian coordinate system, and every cantle of F be presented by a straight line joining the little circles that correspond to its signets. And let the arrow-head be place on the line and indicated the order of the appearance of the signets in a cantle. Also, label every line by the orderate of the cantle. The resulting configuration is in Figure 3.2.1. It is referred to as a plot of F.

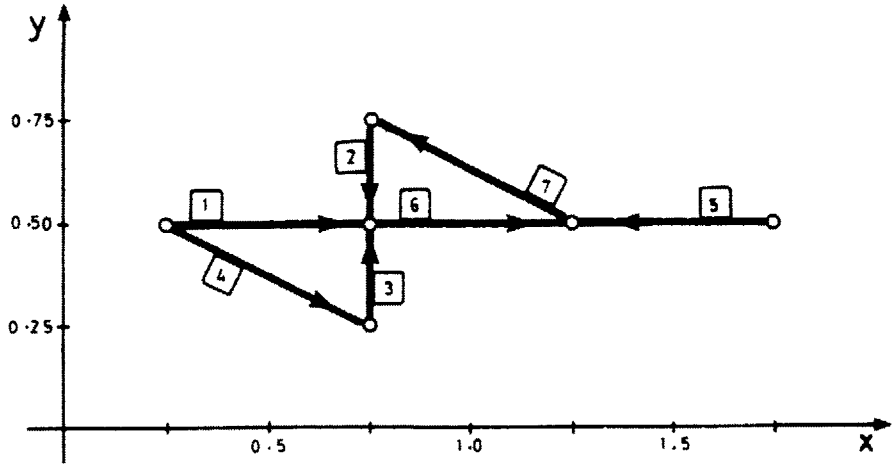


FIGURE 3.2.1: PLOT IN TWO DIMENSIONAL CARTESIAN COORDINATE SYSTEM 1

We can repeat the same procedure but now the center of each circle will be changed due to the function as follows:

$$x = 2\ S_1+S_2-3$$

$$y = \sqrt{3}\ S_2-1$$

The resulting configuration is shown in Figure 3.2.2, which is another plot of

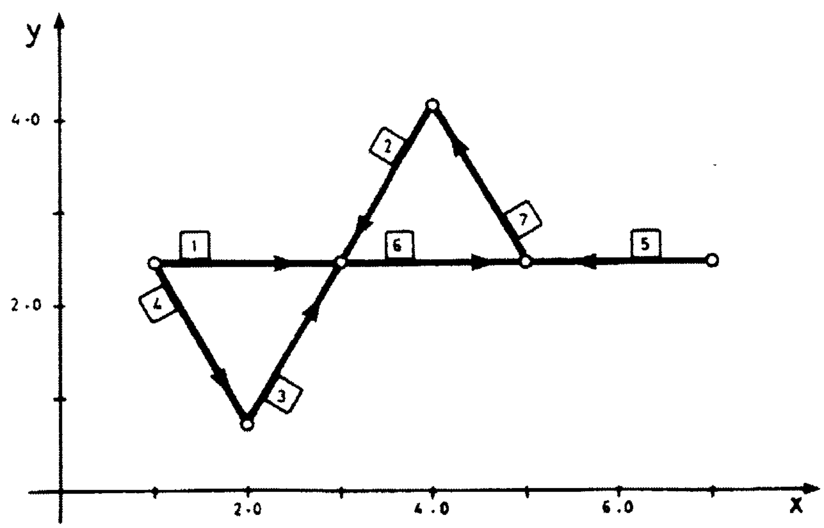


FIGURE 3.2.2: PLOT IN TWO DIMENSIONAL CARTESIAN COORDINATE SYSTEM 2

Now we can repeat the same procedure but using a polar coordinate this time. And the center of each signet will be given by the polar coordinates

$$r = S_1/5$$

$$\theta = (2\ S_2\text{-}1)\pi/10$$

The resulting configuration is shown in Figure 3.2.3.

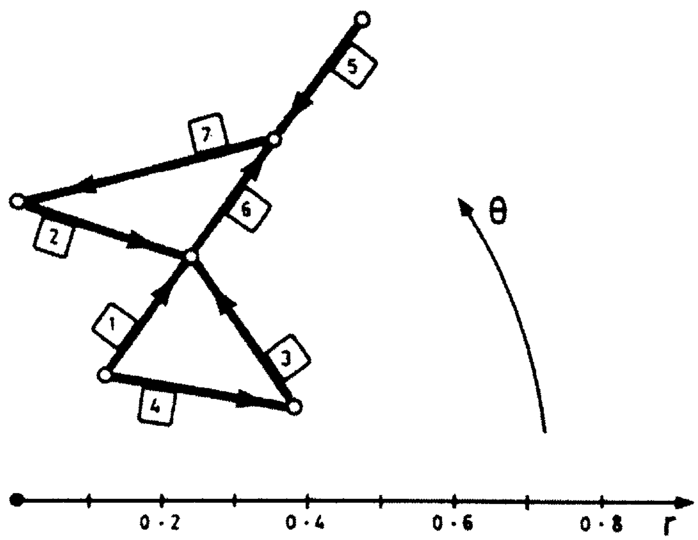


FIGURE 3.2.3 PLOT IN POLAR COORDINATE SYSTEM

CHAPTER FOUR FORMEX FUNCTIONS

4.1 Introduction

In scalar algebra, a relation such as

$$y = x^2 + 2x + 1$$

establishes a rule which is used to evaluate y for any given x. The general equation can be represented by

$$y = f(x)$$

The term f is referred to as a 'function', so y is a function of x. The same rule here applied to the formex functions. In a formex function, a relation such as

$$\mathrm{F} = \Phi \mid \mathrm{G}$$

established a rule which is used to evaluate F for any given G. This rule is representated by a symbol, say Φ. That is, F is defined by G using the equation in Φ. And the symbol '¦' is referred to as the 'rallus symbol' and is read as 'rallus' or 'of'. As one needs to process the same equations repeatedly, it would be more convenient and faster to standardize the process into a function. The name of the formex functions reflects its formex plot. A number of frequently useful formex functions for double-layer grids are described in this chapter.

In discussing the formex functions, the following terminology and notation are used:

(1) If F and G are two formices and

$$\mathrm{F} = \Phi \mid \mathrm{G}$$

then G can be also expressed in terms of F. In this case, the function is said to have an 'inverse'. The inverse of a function Φ is denoted by Φ^{-1} and can be written as

$$G = \Phi^{-1} | F$$

(2) A composite function obtained from repeated application, say m times, of a function Φ is denoted by Φ^{2}. Thus,

$$\Phi | \Phi | G$$

is written as

$$\Phi^{2} | G$$

and

$$\Phi^{-1} | \Phi^{-1} | G$$

is written as

$$\Phi^{-2} | G$$

(3) A composite function that consists of Φ^{m} and Φ^{n} is equivalent to Φ^{m+n}. For example,

$$\Phi^{2} | \Phi^{3} | G = \Phi^{5} | G$$

and

$$\Phi^{8} | \Phi^{-3} | G = \Phi^{5} | G$$

(4) The zeroth power of any function Φ is referred to as an 'identity function' and is expressed as

$$G = \Phi^{0} | G$$

So an identity function maps a formex onto itself.

The functions that are introduced in this thesis belong to the three families of formex functions.

First, there is the family of 'transflection'. There are five basic classes of functions in this family, and they are translation, reflection, vertition, projection and dilatation functions. Any combination of these basic functions is referred to as a transflection.

Secondly, there is the family of 'introflections'. This family consists of three functions, and they are recision, regular variant and absolute recision functions.

Thirdly, there is the family of 'cordations' consisting of four classes of functions that are known as nexum, luxum, conexum and coluxum functions.

In this chapter, translation functions and reflection functions are discussed. All other functions which are not discussed follow the same concept by changing the relationships between F and F'.

4.2 Translation functions

$$F = \{[4, 2, 0; 0, 3, 5], [4, 6, -9], [30, 4, -9]\}$$

and one wants

$$G = \{[4+3, 2,0; 0+3, 3, 5], [4+3, 6, -9], [30+3, 4, -9]\}$$

which means G is a function of F by adding 3 units to the first signet of each cantle of formex F. This relationship can be described in a translation function, and it is written as

$$G = \text{tran}(h,m) \mid F$$

Where h is the direction or the order of the signet, m is the units need to be changed. In this example, h is equal to one and m is equal to 3 since the first signet of each cantle is replaced by 3. Any translation of the empty formex is considered to be the emplty formex itself.

For example, if

$$F1 = \{[1, 2; 2, 2], [2, 1; 2, 2]\}$$

and if

$$F2 = tran(1, 3) ¦ F1,$$

$$F3 = tran(2, 2) ¦ F2$$

and

$$F4 = tran(1, -3) ¦ F3$$

then F2, F3 and F4 are found to be

$$F2 = \{[4, 2; 5, 2], [5, 1; 5, 2]\}$$

$$F3 = \{[4, 4; 5, 4], [5, 3; 5, 4]\}$$

$$F4 = \{[1, 4; 2, 4], [2, 3; 2, 4]\}.$$

The formex plot of F1 to F4 is shown in Figure 4.2.1, where the plot of F_i is denoted by P_i. U1 shows the first direction and U2 shows the second direction of each cantle.

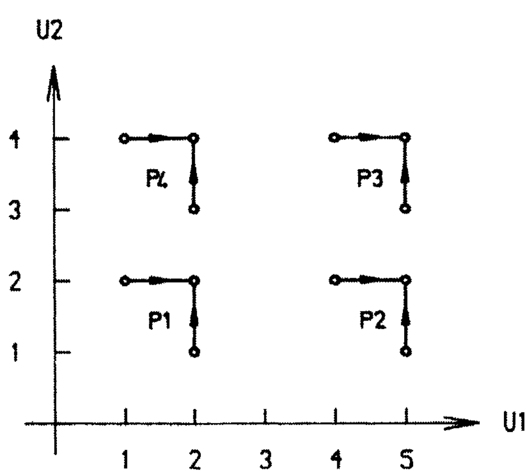


FIGURE 4.2.1: AN EXAMPLE OF FORMEX PLOT OF TRANSLATION FUNCTION 1

Further examples,

F1 = [2, 2; 1, 1],

F2 = [12, 1; 12, 2],

F3 = [12, 9; 11, 10]

and

F4 = [1, 10; 2, 10]

and where F5 to F9 are obtained as

$$F5 = \sum \frac{8}{i=2} tran(1,i) \mid F1,$$

$$F6 = \sum \frac{6}{j=2} tran(2,j) \mid F2,$$

$$F7 = \sum \frac{-2}{j=-8} tran(1,i) \mid F3,$$

$$F8 = \sum \frac{2}{j=7} tran(2,-j) \mid F4$$

and

$$F9 = \sum \frac{3}{j=2} tran(2,j) \mid F5\# \sum \frac{2}{i=9} tran(1,-i) \mid F6\# \sum \frac{-2}{j=-3} tran(2,j) \mid F7\# \sum \frac{8}{i=2} tran(1,i) \mid F8.$$

In order to get F5, F1 is obtained. F1 is called as a 'generant'. Thus, F2, F3 and F4 are the generants of F6, F7 and F8. And F9 is obtained by 4 generants, namely F5, F6, F7 and F8.

The formex plot of F1 to F9 are shown in Figure 4.2.2, where the plot of F_i is denoted by P_i.

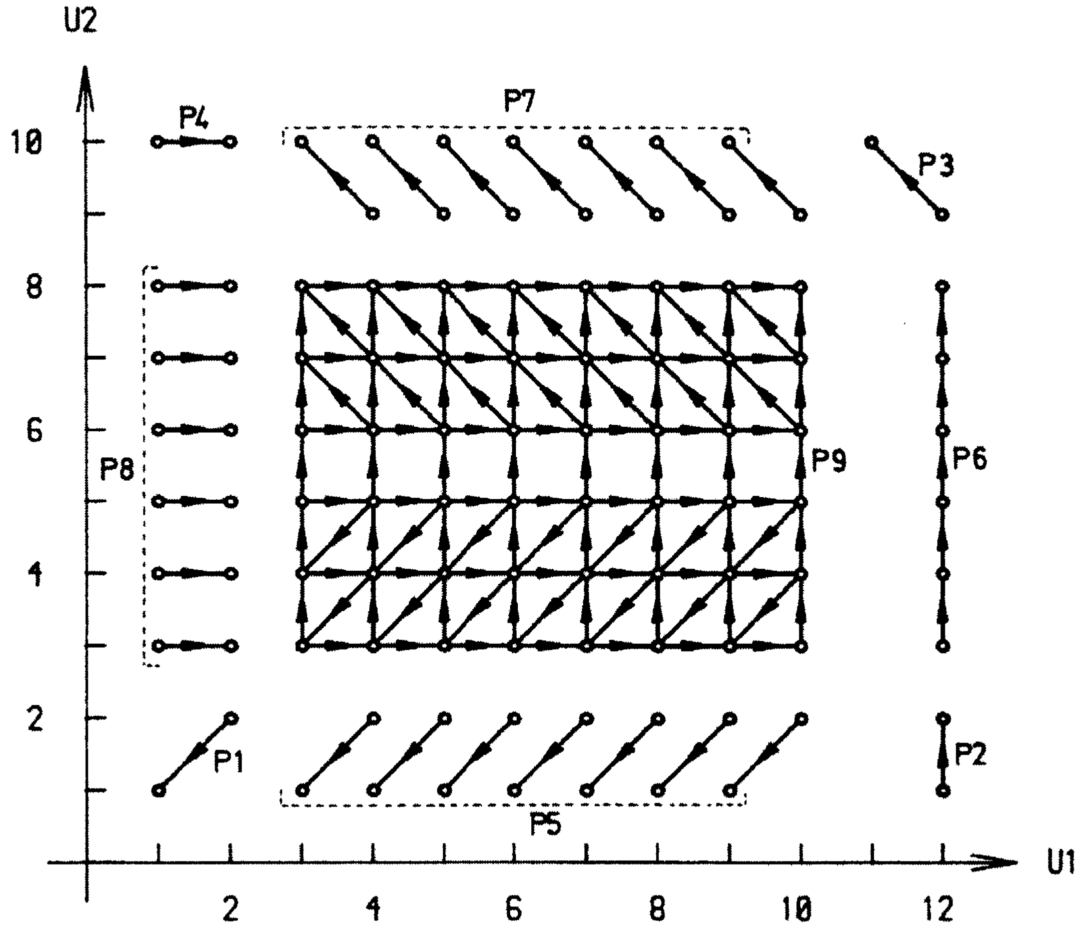


FIGURE 4.2.2: AN EXAMPLE OF FORMEX PLOT OF TRANSLATION FUNCTION2

The relationship between the plots of Figure 4.2.2 are seem to be the same rule described for the plots of Figure 4.2.1. Thus, a translation function tran(h,m) gives rise to a translation by m units in the U_h direction.

The basic properties of translation functions are as the following:

(1) Translation functions are commutative. That is,

$$\text{Tran}(h', u')|\text{tran}(h, u)|F = \text{Tran}(h, u)|\text{tran}(h', u')|F$$

(2) The inverse of a translation function tran(h, u) is the translation function tran(h, - u). That is,

$$\text{tran}(h, u)^{-1}|F = \text{tran}(h, -u)|F$$

(3) tran(h, u')|tran(h, u)|F = tran(h, u+u')|F and this in turn implies that

$$\text{tran}(h, u)^{i}|\ F = \text{tran}(h, iu)|F$$

where i is an integer.

4.3 Reflection functions

F is a formex of rth grade, so F is in the form of $\{c_1, c_2, c_3, c_{4,\ldots\ldots}, c_r\}$. Let F' be a formex of the same grade, F' = $\{c'_1, c'_2, c'_3, c'_{4,\ldots\ldots}, c_r\}$, and every cantle of F' is replaced by every cantle of F where all values of i = 1, 2, 3, 4, ,r except for i = h, so

$$I'_i = I_i,$$

and

$$I'_{h,} = 2u\text{-}I_h$$

Where h is any nonzero integer from 1 to n, u is either an integer or semi-integer, such as 1, 1.5, 2, 2.5, 3, 3.5….

The rule which is described here for F to transform into F' can be symbolized in terms of a formex function. The function is referred as a 'reflection function', and is written as

$$\text{ref}(h, u)$$

So the relationship between F and F' can be written as

$$F' = \text{ref}(h, u)\ |\ F$$

Any reflection of the empty formex is considered to be the empty formex itself.

For example, if

$$F1 = \{[2, 0, -9], [0, 0, 0; 1, 1, 1]$$

and if

$$F2 = ref(2, 2) | F1$$

then F2 is found to be

$$F2 = \{[2, 4, -9], [0, 4, 0; 1, 3, 1]\}$$

The basic properties of the reflection functions are as the following:

(1) Reflection functions that correspond to different directions are commutative.

$$ref(h_1, u_1) | ref(h_2, u_2) | F = ref(h_2, u_2) | ref(h_1, u_1) | F$$

(2) A reflection function is the inverse of itself. That is,

$$ref(h, u)^{-1} | F = ref(h, u) | F$$

This can imply and prove for the following.

$$\begin{aligned} & ref(h, u)^{2w} | F \\ &= ref(h, u)^{w} | F * ref(h, u)^{w} | F \\ &= ref(h, u)^{-w} | F * ref(h, u)^{w} | F \\ &= ref(h, u)^{0} | F \\ &= F \end{aligned}$$

and

$$\begin{aligned} & ref(h, u)^{2w+1} | F \\ &= ref(h, u)^{2w} | F * ref(h, u) | F \\ &= ref(h, u)^{0} | F * ref(h, u) | F \\ &= ref(h, u) | F \end{aligned}$$

So, any even power of the inflection function is equal to an identity function and any odd power of the inflection function is equal to the inflection function itself.

CHAPTER FIVE CASE STUDY

A double-layer grid structure consists of two interconnected parallel networks. As shown in Figure.1, the top grid is indicated by full lines, the bottom grid is indicated by broken lines, and the diagonal bars which connect them are indicated by dotted lines. The structure consists of 280 beam elements that are rigidly connected together at 83 joints. Suppose that the grid is to be subjected to analysis using the standard stiffness method in conjunction with a computer program and that it is necessary to prepare the necessary input data. For this purpose, one has to provide information about the interconnection pattern, geometric particulars, material properties, external loads and support conditions.

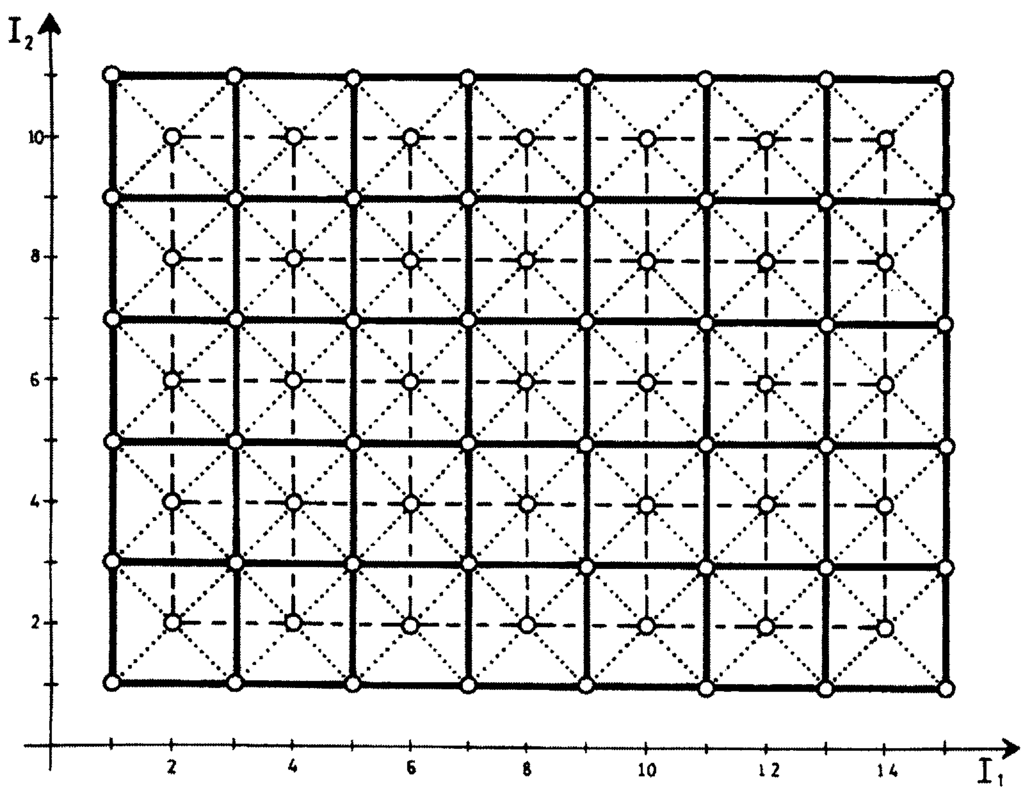


FIGURE 5.1: NORMAL Z-PLOT

A major part of the data consists of a description of the interconnection pattern of the structure, thus we focus one attention on this particular aspect of data preparation.

The interconnection pattern of the structure may be described using the concepts of formex algebra. Consider only the top grid for now. Focusing on the left bottom corner of the top grid, an element such as e_1 can be described by the cantle

$$[1, 1; 3, 1]$$

This specifies that joint j_1 is connected to joint j_2. Similarly, element e_2 can be described by the cantle

$$[3, 1; 5, 1]$$

The combination of element e_1 and e_2 can be represented by the construct

$$\{[1, 1; 3, 1], [3, 1; 5,1]\}$$

which is an example of a formex.

Using the method described above, one may now describe each individual element or their combinations in terms of formices. For example,

$$F_1 = [1, 1; 3, 1]$$

and

$$F_2 = [3, 1; 5, 1]$$

represent the elements e_1 and e_2, respectively, and

$$F_{12} = \{[1, 1; 3, 1], [3, 1; 5, 1]\}$$

represent the pairs of elements e_1 and e_2.

To obtain a formex representing the combination of the elements e_1 and e_2, one may also write

$$F_{12} = F_1 \# F_2$$

which is an example of formex composition.

More example,

$$F_3 = [1, 5; 1, 7]$$

and

$$F_4 = [1, 7; 1, 9]$$

represent the element e_2 e_3 and e_4, respectively.

If

$$F = F_1 \# F_2 \# F_3 \# F_4$$

then this may be written as

$$\sum \frac{4}{i=1} F_i$$

which is an example of libra construction.

The above formex F is written as

$$F = \{[1, 1; 3, 1], [3, 1; 5, 1], [1, 5; 1, 7], [1, 7; 1, 9]\}$$

Another way to show F is to use the 'formex function'.

$$\mathrm{tran}(1, 1) \mid$$

is a 'formex function' implying 'translation' in the first direction by 1 unit.

The symbol ' ¦ ' is called the 'rallus symbol' and has the role of separating the function from its argument. For instance,

$$F_2 = \mathrm{tran}(1, 2) \mid F_1$$

$$F_3 = \mathrm{tran}(1, 2) \mid F_1$$

and

$$F_4 = \mathrm{tran}(1, 2) \mid F_3$$

Now, let it be required to write a formex representing the combination of F_1, F_2, F_3, F_4, F_5, F_6 and F_7 (representing e_1, e_2, e_3, e_4, e_5, e_6 and e_7).

$$\sum \frac{6}{i=0} tran(1,2i) \mid F$$

where F = [1, 1; 3, 1].

Translation of F by zero unit in the first direction is F itself. That is

tran(1, 0)| F =F.

A translation function may imply translation not only in the first direction but also in the second direction. For instance, now the top layer grid parallel to the I_1 axis maybe described by the formex

$$E_1 = \sum \frac{5}{j=0} tran(2,2j) \mid \sum \frac{6}{i=0} tran(1,2i) \mid F$$

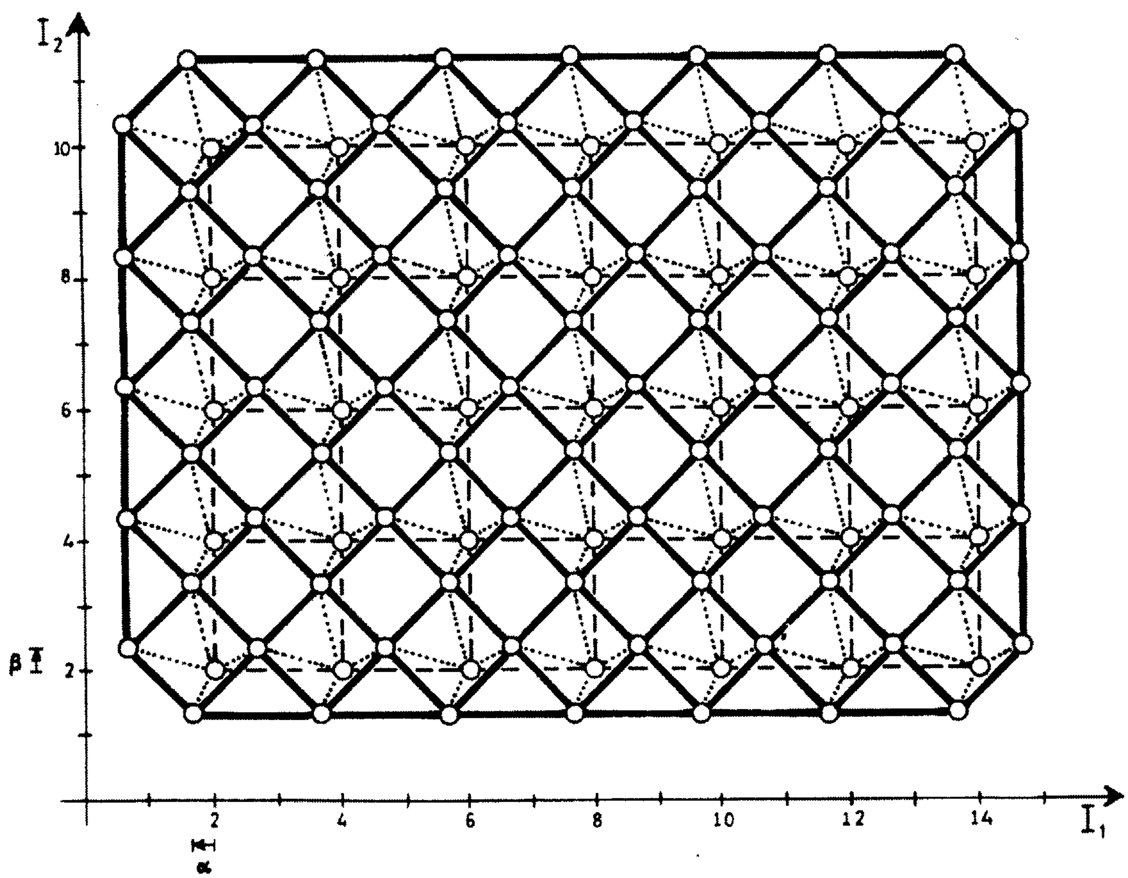
where F is equal to [1, 1; 3, 1], and the top layer grid parallel to the I_2 axis maybe described by the formex

$$E_2 = \sum \frac{7}{i=0} tran(1,2i) \mid \sum \frac{4}{j=0} tran(2,2j) \mid G$$

where G is equal to [1, 1; 1, 3].

Let us consider now the third direction of the cantle. The interconnection pattern of a double-layer grid is represented by a homogeneous formex of the third grade and second plexitude in which each cantle represents an element of the grid. Furthermore, the bottom grid elements lie in the I_1-I_2 plane and the top grid elements lie in a plane which is parallel to the I_1-I_2 plane and intersects the I_3 axis at $I_3 = 1$. So the third index of a signet in the formex of the top grid is always equal to 1, and 0 for the bottom grid.

Formex plots that represent double-layer grids can be shown in the style of Figure 5.1 or Figure 5.2. Figure 5.1 is called as a 'normal Z-plot', and Figure 5.2 is called as an 'oblique Z-plot'. An oblique Z-plot is used when some of the elements of the grids are coincides in its normal Z-plot. The term Z-plot is used to refer to either a normal Z-plot or an obliqueZ-plot.



FIGRUE 5.2: OBLIQUE Z-PLOT

Now E_1 can be described by

$$E_1 = \sum_{i=0}^{6} \sum_{j=0}^{5} tranid(2i,2j) \mid 1,1,1;3,1,1]$$

E_2 can be described by

$$E_2 = \sum_{i=0}^{7} \sum_{j=o}^{4} tranid(2i, 2j) \mid [1,1,1;1,3,1]$$

Furthermore, the bottom layer grid, which is parallel to the I_1 axis can be described by

$$E_3 = \sum_{i=0}^{5} \sum_{j=0}^{4} tranid(2i, 2j) \mid [2,2,0;4,2,0]$$

And the bottom layer grid, which is parallel to the I_2 axis can be described by

$$E_4 = \sum_{i=0}^{6} \sum_{j=0}^{3} tranid(2i, 2j) \mid [2,2,0;2,4,0]$$

CHAPTER SIX   STATIC ANALYSES OF DOUBLE-LAYER GRIDS

6.1 Geometry

Space structures are typical examples of skeleton frameworks. They consist of a large number of simple modular, prefabricated units, often of standard size and shape, which combine into a light, but very rigid, three-dimensional structure. In order to have standard size and shape units, the geometry of the double-layer grid roof needs to be carefully determined.

The double-layer grids in this thesis are based on squares. The top grid is offset from the bottom grid in plan but has the same direction. The grids are connected by diagonal members. The square is 1 by 1 meter and the diagonal is also 1 meter.

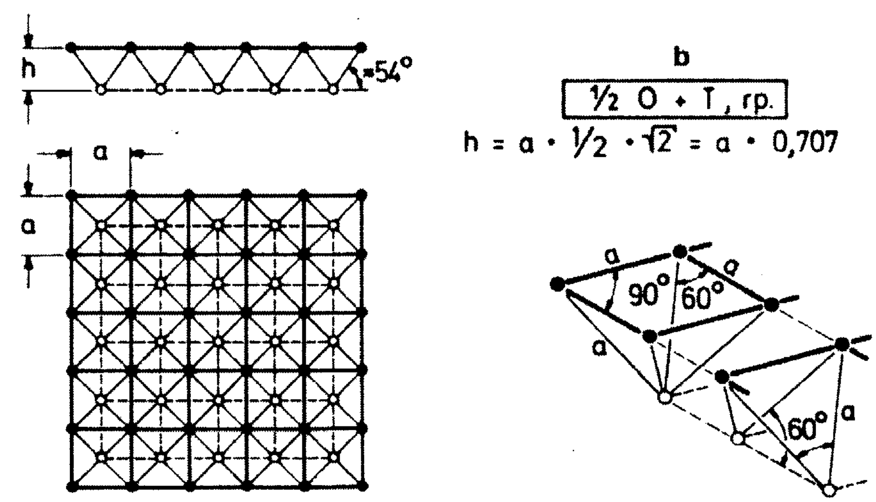


FIGURE 6.1: GRID GEOMETRY

6.2 Structural Analysis

The distributed load of 30 psf(equal 1.4KN per square meter) is applied as point loads at each joint of the top grid. The structure is supported at its four corners. The loads are 0.35 KN at the four corners; 0.7 KN is at the other joints along the sides; 1.4 KN placed at the rest of the joints on the top grid. All elements have only axial loads since the structure is considered to be a space truss.

In the finite element method, truss elements can be represented as 2-node, 3-node and 4-node elements. In this example, a 2-node element was employed since 2-node elements are adequate for modeling a truss. When the material is linear elastic, the 2-node truss element requires only 1-point Gauss numerical integration for an exact evaluation of the stiffness matrix, since the strain is constant in the element. In this example, steel pipes were used as the structural members. Analyses were carried out using 2 programs, ADINA and SAP.

The double-layer grid structure is supported at the four corners with hinges that constrain translation. In this example, one should compare the forceRR stressRR and strainRR of each element and the displacement of each joint. Since stress RR is equal the product of the inverse of the cross sectional area and the force, we will consider only force and displacement here. This also applies to strainRR.

The first example is a 5 by 5 meter double-layer grid. When we model it in ADINA and SAP, we find that the forces in each element are very similar. Since the structure is symmetric, we expect the answers to be the same. We do not have to make the calculation by hand using the method of joints, since it is time consuming and not very efficient[1].

The second example is a 2 by 2 meters double-layer grid. We could use the method of joints to find the force in each element. If we compare the forces from the method of the joints and ADINA, we observe that they are almost the same[2].

6.3 Numerical solution

In the case of the 2 by 2 grid, the elements on the side carry zero force. The diagonals that connect directly to the support are subjected to tension. The rest of the diagonals are subjected to compression. The elements on the bottom grid are subjected to tension.

---

[1] For calculation output, refer to appendix A

[2] For calculation output, refer to appendix B

In the case of the 5 by 5 grid, the elements on the side carry small compression. The diagonals that connect directly to the support carry extremely high tensile force. On the bottom grid, the side elements carry the second highest tension of the structure. The tensile force decreases with distance from the edge toward the interior. With respect to translation, the inner joints always have more deflection than the outer joints.

CHAPTER SEVEN DESIGN OF A SPACE TRUSS ROOF

7.1 Geometry

The elevation view of the space truss is an arc with a radius of 80 feet. By dividing the angle of the arc, we can cut the arc into many equal length units. Basically the double-layer grids are based on squares. The top grid is offset from the bottom grid in plan but remaining directionally the same. The grids are connected by diagonal members. The square is 3 by 3 feet and the diagonal is also 3 feet.

7.2 Structural Analysis

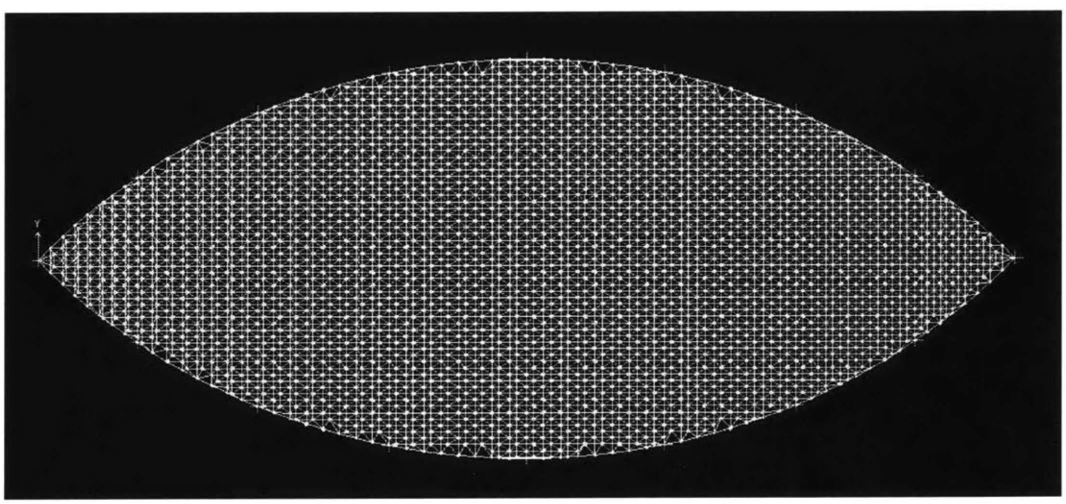


FIGURE 7.2.1: PLAN VIEW OF THE STRUCTURE IN SAP

There are more than ten thousand elements in each roof-truss, and because of the complex curvature of the roof, we had to draw them first on CAD, and then export the result to SAP. Gravity loads were placed at each joint of the top grid. All elements have only axial loads going through. The roof is supported by the steel columns, which support the whole structure. These steel columns take the diaphragm loads. After analyzing the structure, a 3-inch pipe with the thickness of 0.6 inches was chosen for the columns. A few members have irregular length. But more than 30 thousand elements will be produced in the same size making the design very economical[3].

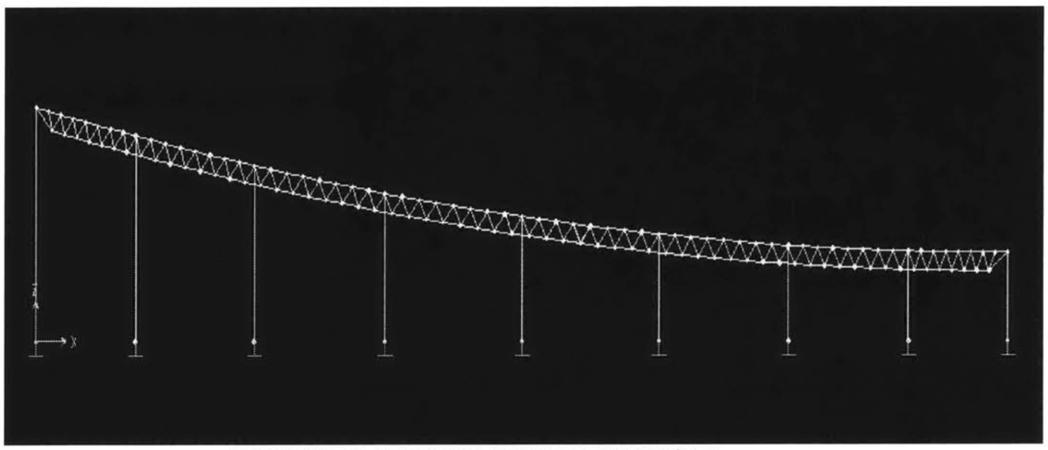


FIGURE 7.2.2: ELEVATION VIEW IN SAP

7.3 Mero System

Mero was invented by Dr. Max Mengeringhausen. The principle of the system is the prefabrication and standardization of the element, which makes Mero economical. Standard Mero connectors have 18 screw holes. This connector has 18 surfaces connecting angles of 45, 60, 90 and multiples of these angles. It may be made in different sizes to resist different forces. The regular Mero connector has only 10

[3] For calculation output, refer to appendix C.

faces, but special Mero connectors can be made, with holes drilled at special angles to suit particular applications. The minimum angle between two adjacent holes is 35 degrees.

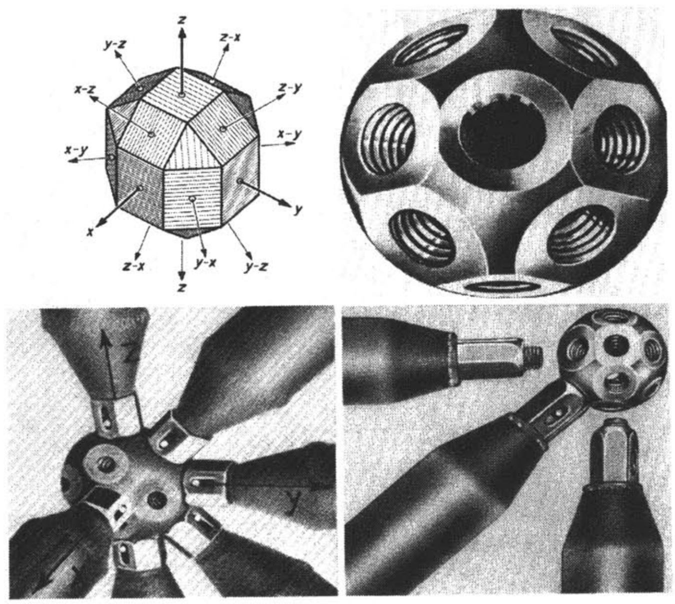


FIGURE 7.3.1: A STANDARD MERO CONNECTOR

The issue of safety in a fire is important when using a space truss. In case of fire, collapse of double-layer grids is commonly characterized by the sequential failure of individual members. As fire develops, temperatures increase and the progressive weakness of steel causes failure of members. After collapse of some parts of the structure, the remaining sections are stable and still able to resist the applied load.

The erection of the Mero space system does not produce any special problems. Basically, all the usual erection techniques and problems encountered during the assembly of steel structures obviously also apply to the Mero space structures. A mixed erection technique was considered in our case. Large parts of the structure are first interconnected at ground level, and then, with the help of simple lifting gear, hoisted up into their final position and connected together, forming the final structure. The large amount of labor required for normal assembly will be reduced through the erection of sizable parts at ground level. Additionally, the usual high cost of renting the hoisting equipment is minimized, as the cranes are used only when their load-carrying capacity can be exploited to the fullest, to lift up very bulky or heavy parts.

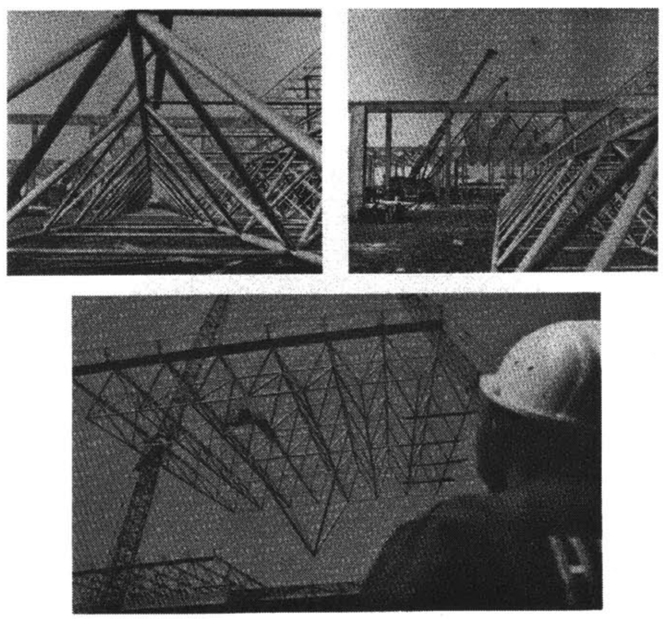


FIGURE 7.3.2: AN EXAMPLE OF THE MIXED ERECTION TECHNIQUE

CHAPTER EIGHT CONCLUSION

Over the last century, engineers have been striving to build larger structures with less material and less cost. Space structures have the incredible rigidity and ability to cover large spans with minimum weight. Double-layer grids belong to the family of space structure. The complex and regular nature of double-layer structural grids require the use of computer techniques for the design and production.

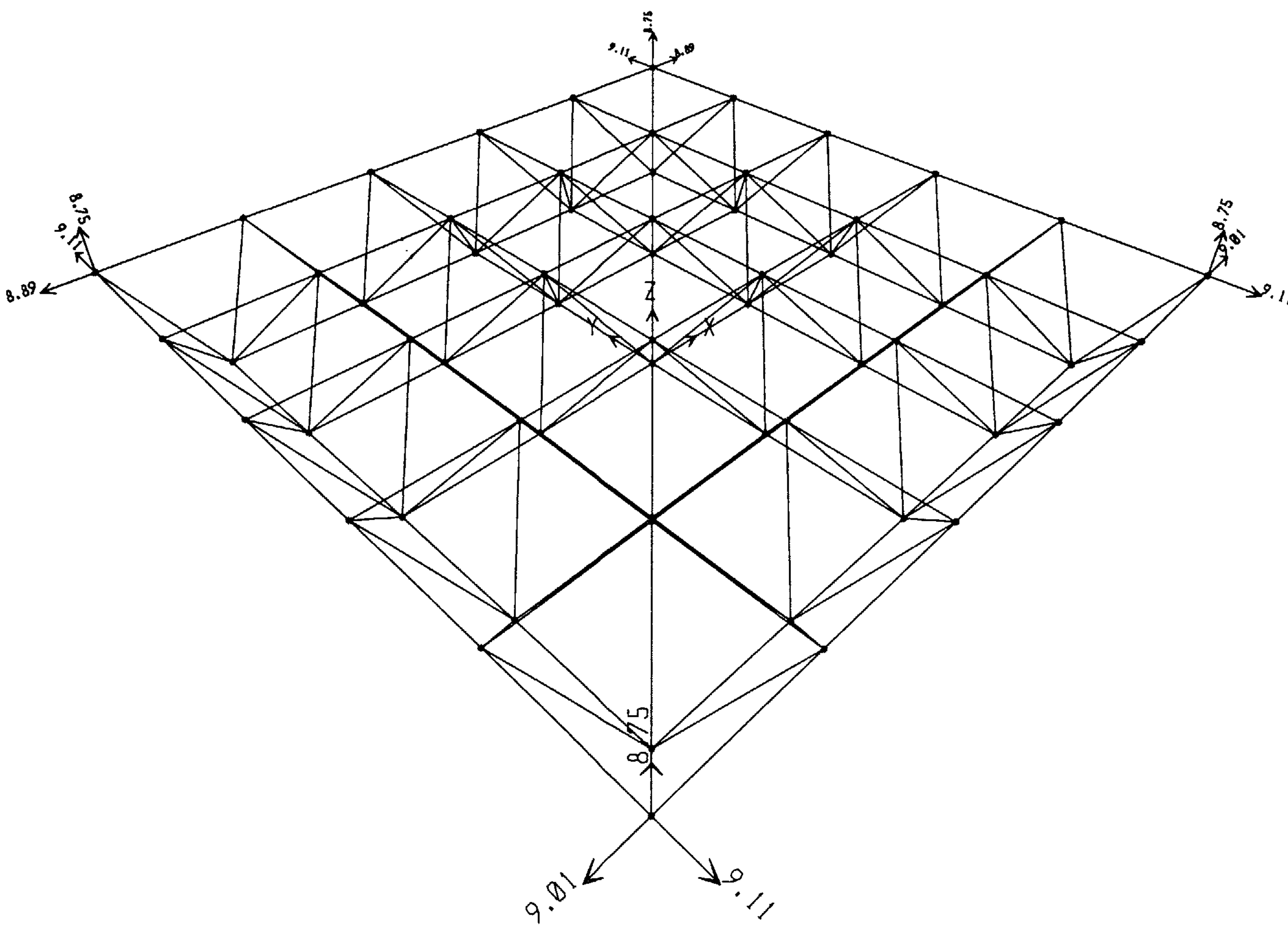
When a structural system is to be analyzed by a digital computer, it is necessary to provide a complete description of the system in an appropriate form. This constitutes the data that are to be used in conjunction with a program. Data preparation is normally a straightforward process involving nothing more than a systematic recording of known facts. For large and complex systems, it is more difficult to control the input data.

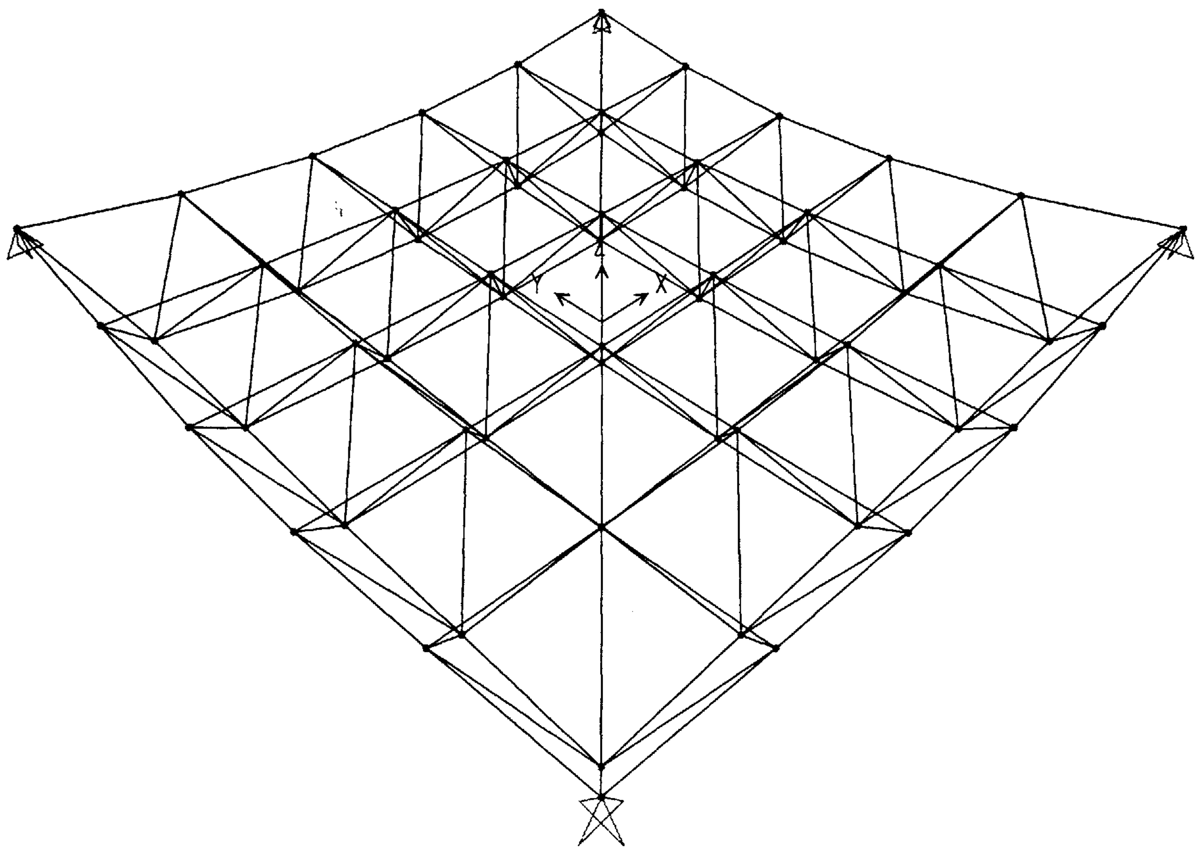
Formex algebra may be used in different ways to overcome the difficulties of data preparation. In particular, the description of the interconnection pattern of a structural system may be conveniently formulated through the concepts of formex algebra.

Formex plots and formex functions are a set of rules extended from formex algebra to minimize the work of data preparation. The concept of formex algebra can be applied not only in double-layer grids but also in other engineering disciplines since it is a mathematical system. This thesis demonstrated how the input data of a complex geometry form, consisting of double-layer grids and described by many nodes, can be reduced to only one equation by using formex formulation.

The last two chapters of the thesis focused on the analysis of double-layer grids. A double-layer grids is simply a three dimensional truss structure. The principle of a truss structure is that each element takes only axial loads. Each element of the double-layer grid carries load differently depending on the support condition of the entire structure. A square based double-layer grid supported on its four corners is the special case. The sides of the structure carry relatively small compression. The diagonals that connect directly to the support carry the highest tensile force. And the further the element is from the support, the higher the deflection it is subjected to.

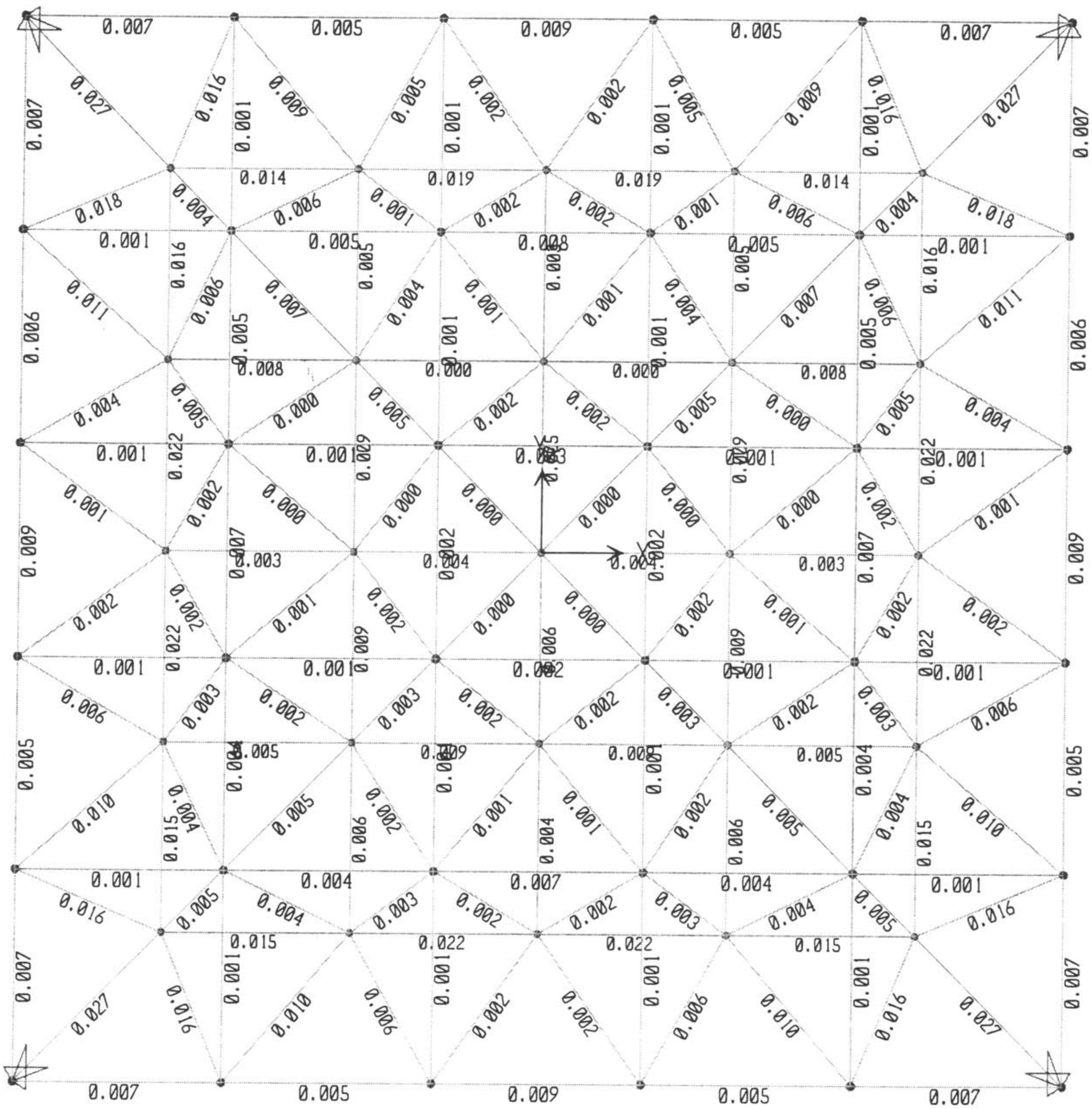
APPENDIX A


8.75
9.11
8.89
Z
Y
X
9.11
9.01
8.75
9.11
8.75
9.11
8.89
8.75
9.01
9.11



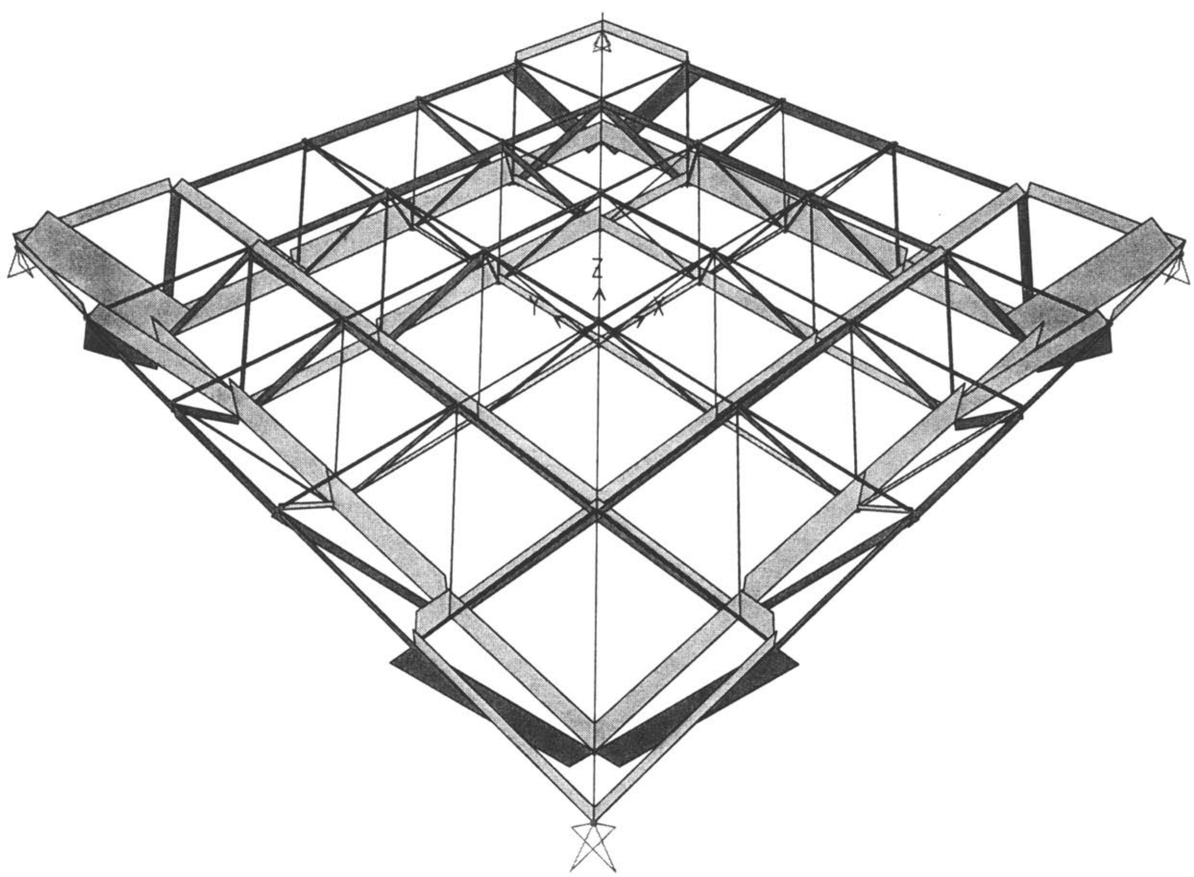
SAP2000 v6.13 - File:fem - Deformed Shape (LOAD1) - KN-m Units

SAP2000 May 14,1999 11:52

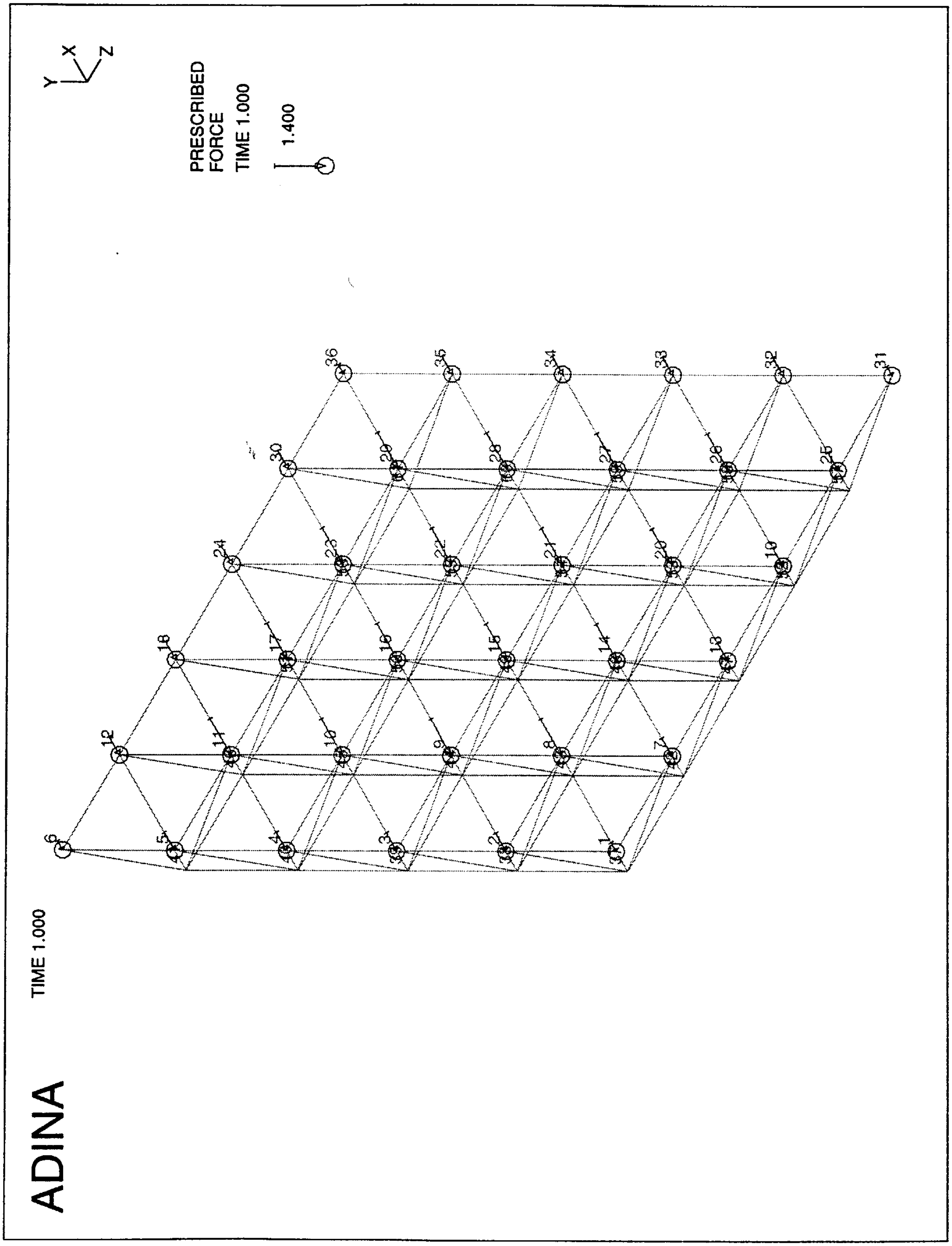


0.00 0.50 0.70 0.90 1.00

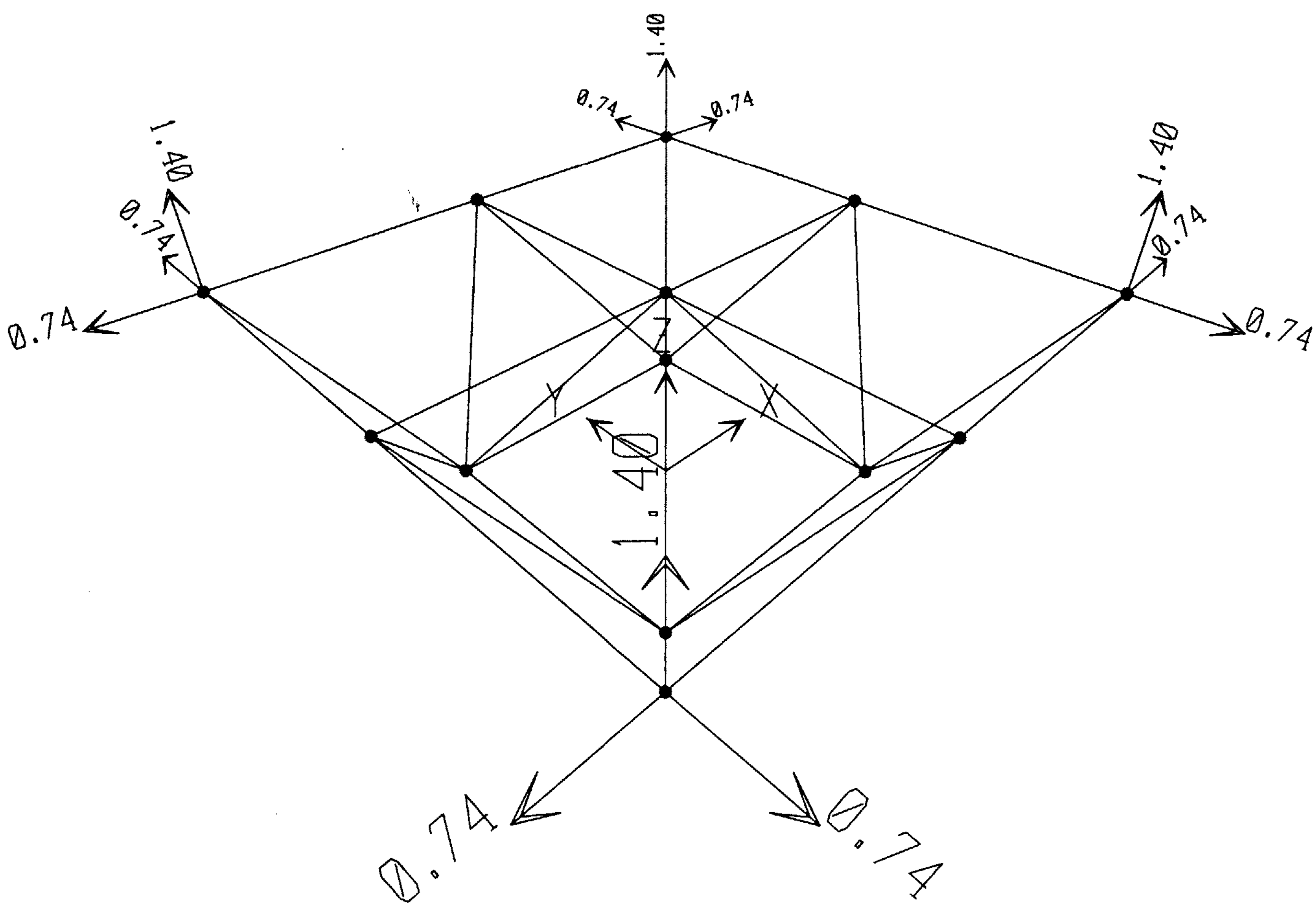
SAP2000 v6.13 - File:fem - Steel P-M Interaction Ratios (AISC-ASD89) - KN-m Units



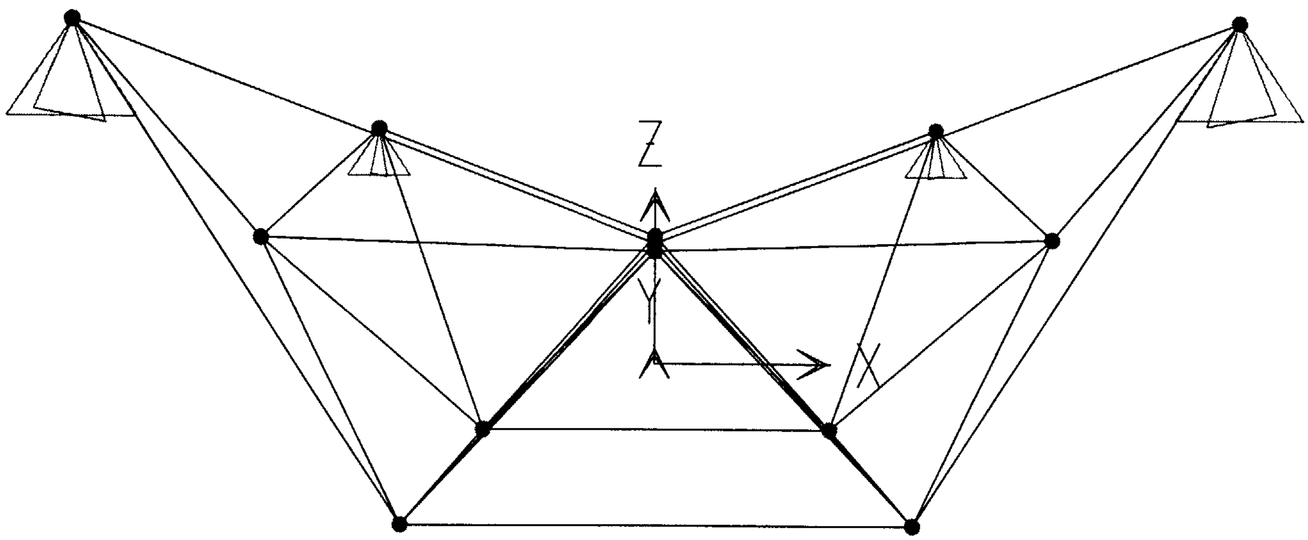
SAP2000 v6.13 - File:fem - Axial Force Diagram (LOAD1) - KN-m Units


ADINA
TIME 1.000
PRESCRIBED
FORCE
TIME 1.000
1.400
Y
X
Z

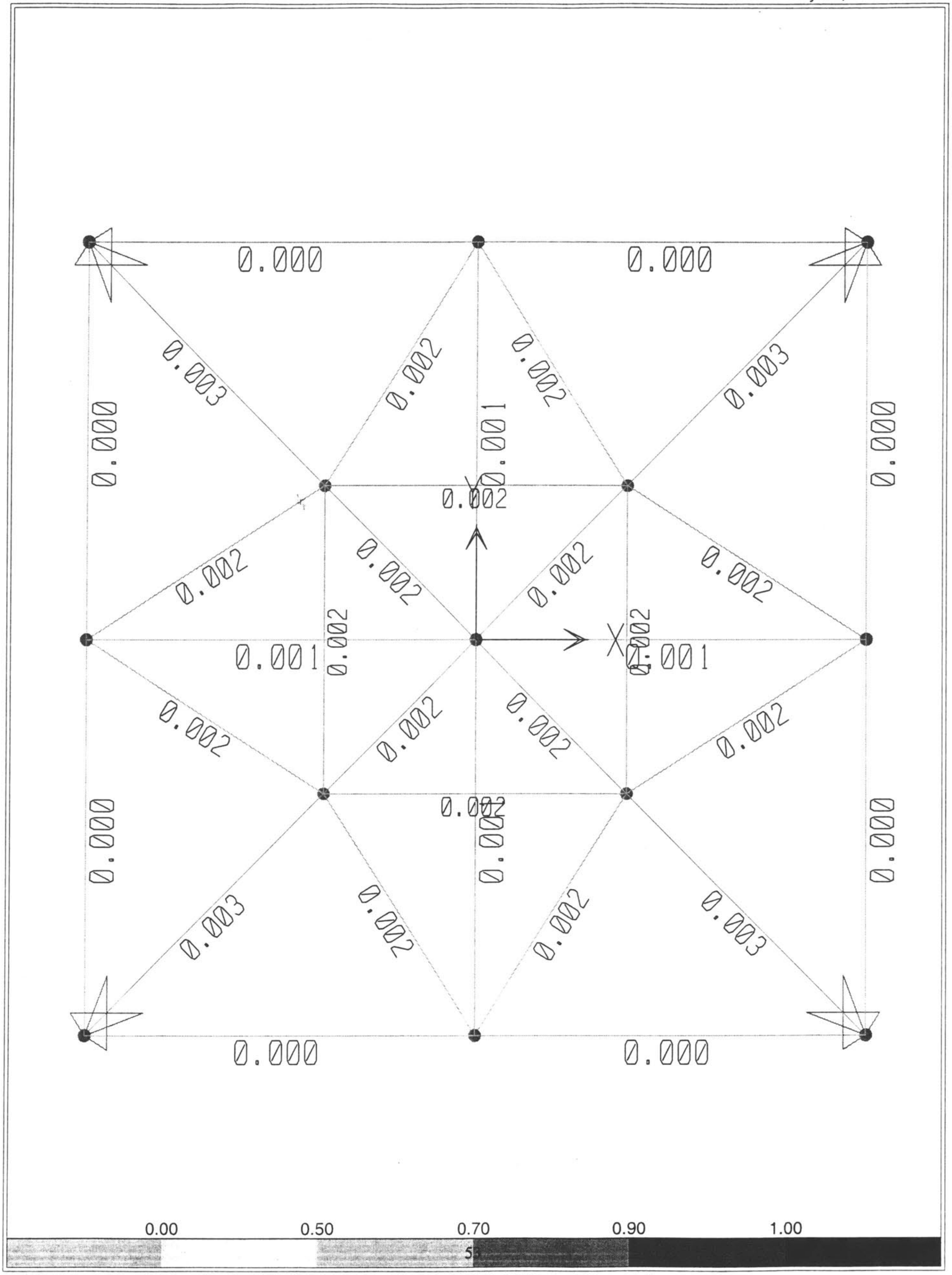
APPENDIX B



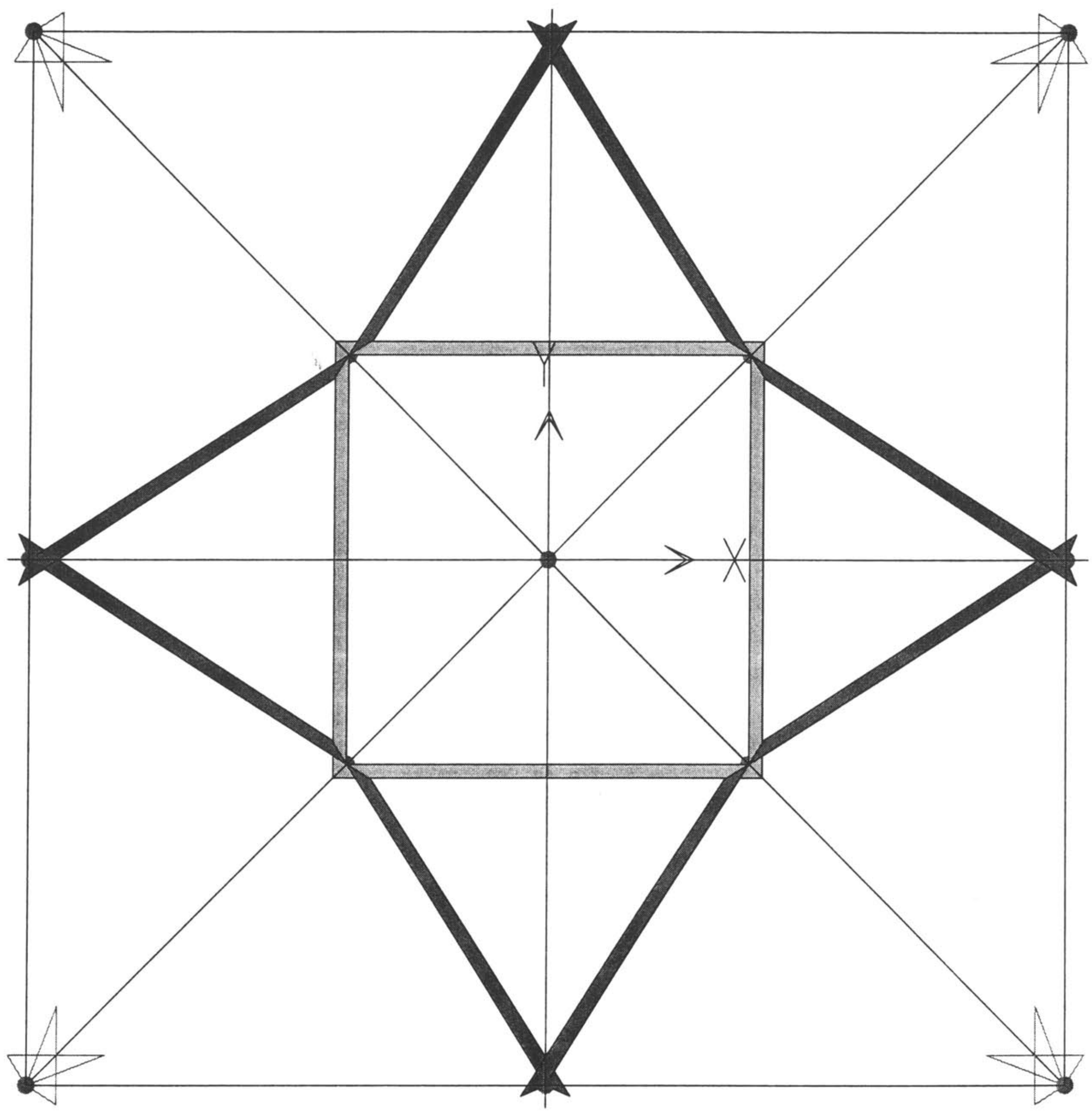
SAP2000 v6.13 - File:fem2 - Restraint Reactions (LOAD1) - KN-m Units



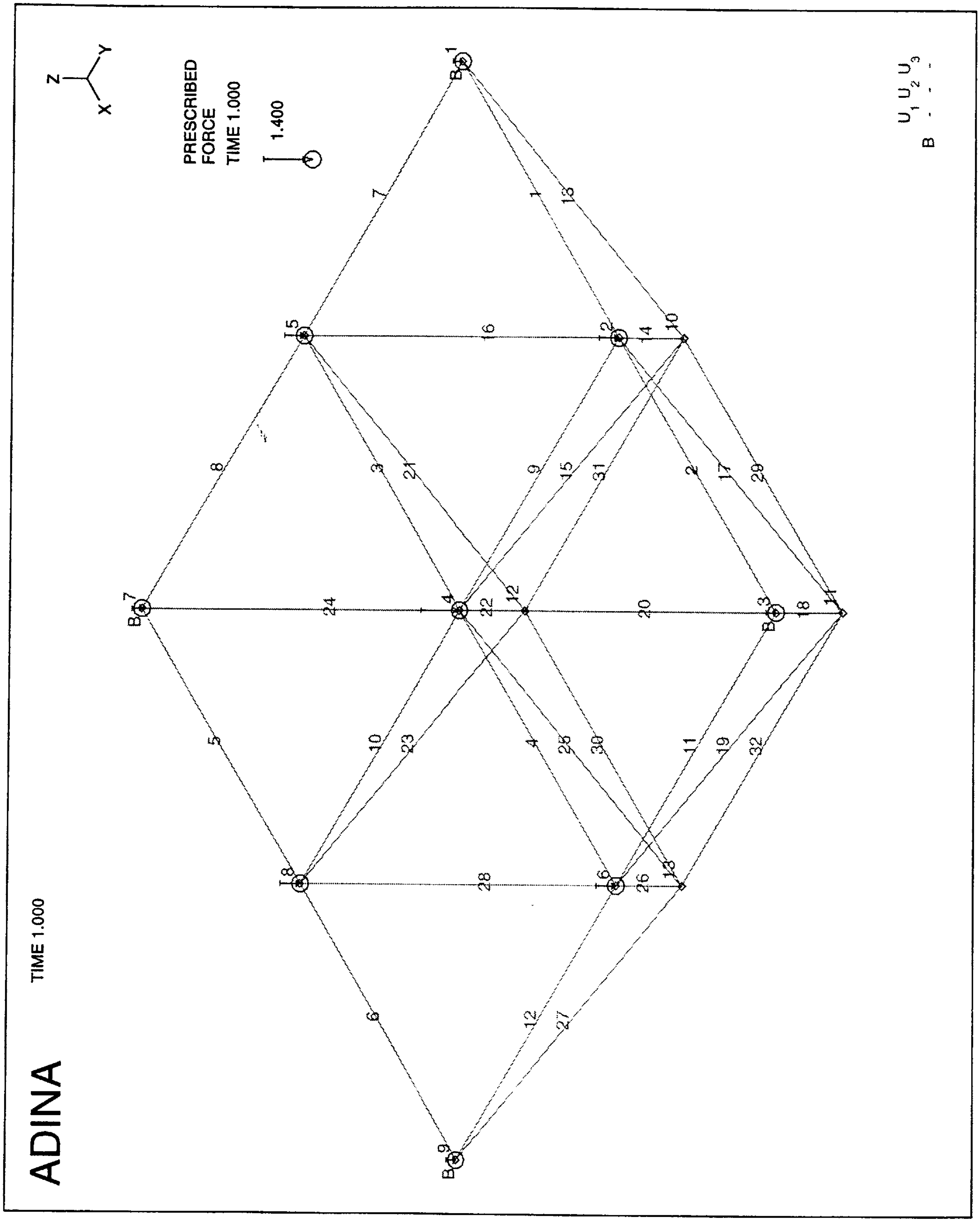
SAP2000 v6.13 - File:fem2 - Deformed Shape (LOAD1) - KN-m Units



SAP2000 v6.13 - File:fem2 - Steel P-M Interaction Ratios (AISC-ASD89) - KN-m Units



SAP2000 v6.13 - File:fem2 - Axial Force Diagram (LOAD1) - KN-m Units


ADINA
TIME 1.000
PRESCRIBED
FORCE
TIME 1.000
1.400
X
Y
Z
U_1 U_2 U_3
B - - -

```
ADINA: AUI version 7.2.2, 14 May 1999:  *** NO HEADING DEFINED ***
Licensed from ADINA R&D, Inc.
Finite element program ADINA, response range type load-step:
Listing for zone WHOLE_MODEL:
     POINT        X-DISPLACEMENT

Time 0.00000E+00

Node 1                  0.00000E+00
Node 2                  0.00000E+00
Node 3                  0.00000E+00
Node 4                  0.00000E+00
Node 5                  0.00000E+00
Node 6                  0.00000E+00
Node 7                  0.00000E+00
Node 8                  0.00000E+00
Node 9                  0.00000E+00
Node 10                 0.00000E+00
Node 11                 0.00000E+00
Node 12                 0.00000E+00
Node 13                 0.00000E+00

Time 1.00000E+00

Node 1                  0.00000E+00
Node 2                  1.43386E-22
Node 3                  0.00000E+00
Node 4                  3.38813E-21
Node 5                 -2.35738E-06
Node 6                  2.35738E-06
Node 7                  0.00000E+00
Node 8                  6.35275E-22
Node 9                  0.00000E+00
Node 10                -2.35738E-06
Node 11                 2.35738E-06
Node 12                -2.35738E-06
Node 13                 2.35738E-06


*** End of list.
```

```
ADINA: AUI version 7.2.2, 14 May 1999:  *** NO HEADING DEFINED ***
Licensed from ADINA R&D, Inc.
Finite element program ADINA, response range type load-step:
Listing for zone WHOLE_MODEL:
     POINT         Y-DISPLACEMENT

Time 0.00000E+00

Node 1              0.00000E+00
Node 2              0.00000E+00
Node 3              0.00000E+00
Node 4              0.00000E+00
Node 5              0.00000E+00
Node 6              0.00000E+00
Node 7              0.00000E+00
Node 8              0.00000E+00
Node 9              0.00000E+00
Node 10             0.00000E+00
Node 11             0.00000E+00
Node 12             0.00000E+00
Node 13             0.00000E+00

Time 1.00000E+00

Node 1              0.00000E+00
Node 2              2.35738E-06
Node 3              0.00000E+00
Node 4              6.35275E-21
Node 5              6.35275E-22
Node 6              6.35275E-22
Node 7              0.00000E+00
Node 8             -2.35738E-06
Node 9              0.00000E+00
Node 10             2.35738E-06
Node 11             2.35738E-06
Node 12            -2.35738E-06
Node 13            -2.35738E-06

*** End of list.
```

```
ADINA: AUI version 7.2.2, 14 May 1999:  *** NO HEADING DEFINED ***
Licensed from ADINA R&D, Inc.
Finite element program ADINA, response range type load-step:
Listing for zone WHOLE_MODEL:
     POINT        Z-DISPLACEMENT

Time 0.00000E+00

Node 1                0.00000E+00
Node 2                0.00000E+00
Node 3                0.00000E+00
Node 4                0.00000E+00
Node 5                0.00000E+00
Node 6                0.00000E+00
Node 7                0.00000E+00
Node 8                0.00000E+00
Node 9                0.00000E+00
Node 10               0.00000E+00
Node 11               0.00000E+00
Node 12               0.00000E+00
Node 13               0.00000E+00

Time 1.00000E+00

Node 1                0.00000E+00
Node 2               -1.83359E-05
Node 3                0.00000E+00
Node 4               -2.00030E-05
Node 5               -1.83359E-05
Node 6               -1.83359E-05
Node 7                0.00000E+00
Node 8               -1.83359E-05
Node 9                0.00000E+00
Node 10              -1.33351E-05
Node 11              -1.33351E-05
Node 12              -1.33351E-05
Node 13              -1.33351E-05

*** End of list.
```

```
ADINA: AUI version 7.2.2, 14 May 1999:  *** NO HEADING DEFINED ***
Licensed from ADINA R&D, Inc.
Finite element program ADINA, response range type load-step:
Listing for zone WHOLE_MODEL:
Element field variables are evaluated using RST interpolation.
      POINT              STRESS-RR

Time 1.00000E+00

Element 1 of element group 1

Int point 1             1.00370E-14

Element 2 of element group 1

Int point 1            -1.00370E-14

Element 3 of element group 1

Int point 1             1.65017E+02

Element 4 of element group 1

Int point 1             1.65017E+02

Element 5 of element group 1

Int point 1             4.44692E-14

Element 6 of element group 1

Int point 1            -4.44692E-14

Element 7 of element group 1

Int point 1            -4.44692E-14

Element 8 of element group 1

Int point 1             4.44692E-14

Element 9 of element group 1

Int point 1             1.65017E+02

Element 10 of element group 1

Int point 1             1.65017E+02

Element 11 of element group 1

Int point 1            -4.44692E-14

Element 12 of element group 1

Int point 1             4.44692E-14

Element 13 of element group 1

Int point 1             4.95012E+02
```



```
Element 14 of element group 1
```

```
Int point 1             -1.65004E+02
Element 15 of element group 1
Int point 1             -1.65004E+02
Element 16 of element group 1
Int point 1             -1.65004E+02
Element 17 of element group 1
Int point 1             -1.65004E+02
Element 18 of element group 1
Int point 1              4.95012E+02
Element 19 of element group 1
Int point 1             -1.65004E+02
Element 20 of element group 1
Int point 1             -1.65004E+02
Element 21 of element group 1
Int point 1             -1.65004E+02
Element 22 of element group 1
Int point 1             -1.65004E+02
Element 23 of element group 1
Int point 1             -1.65004E+02
Element 24 of element group 1
Int point 1              4.95012E+02
Element 25 of element group 1
Int point 1             -1.65004E+02
Element 26 of element group 1
Int point 1             -1.65004E+02
Element 27 of element group 1
Int point 1              4.95012E+02
Element 28 of element group 1
Int point 1             -1.65004E+02
Element 29 of element group 1
Int point 1              3.30033E+02
Element 30 of element group 1
Int point 1              3.30033E+02
```

```
Element 31 of element group 1

Int point 1               3.30033E+02

Element 32 of element group 1

Int point 1               3.30033E+02


*** End of list.
```

ADINA: AUI version 7.2.2, 14 May 1999: *** NO HEADING DEFINED ***
Licensed from ADINA R&D, Inc.
Finite element program ADINA, response range type load-step:
Listing for zone WHOLE_MODEL:
Element field variables are evaluated using RST interpolation.

POINT	STRAIN-RR
Time 1.00000E+00	
Element 1 of element group 1	
Int point 1	1.43386E-22
Element 2 of element group 1	
Int point 1	-1.43386E-22
Element 3 of element group 1	
Int point 1	2.35738E-06
Element 4 of element group 1	
Int point 1	2.35738E-06
Element 5 of element group 1	
Int point 1	6.35275E-22
Element 6 of element group 1	
Int point 1	-6.35275E-22
Element 7 of element group 1	
Int point 1	-6.35275E-22
Element 8 of element group 1	
Int point 1	6.35275E-22
Element 9 of element group 1	
Int point 1	2.35738E-06
Element 10 of element group 1	
Int point 1	2.35738E-06
Element 11 of element group 1	
Int point 1	-6.35275E-22
Element 12 of element group 1	
Int point 1	6.35275E-22
Element 13 of element group 1	
Int point 1	7.07160E-06
Element 14 of element group 1	

```
Int point 1            -2.35720E-06
Element 15 of element group 1
Int point 1            -2.35720E-06
Element 16 of element group 1
Int point 1            -2.35720E-06
Element 17 of element group 1
Int point 1            -2.35720E-06
Element 18 of element group 1
Int point 1             7.07160E-06
Element 19 of element group 1
Int point 1            -2.35720E-06
Element 20 of element group 1
Int point 1            -2.35720E-06
Element 21 of element group 1
Int point 1            -2.35720E-06
Element 22 of element group 1
Int point 1            -2.35720E-06
Element 23 of element group 1
Int point 1            -2.35720E-06
Element 24 of element group 1
Int point 1             7.07160E-06
Element 25 of element group 1
Int point 1            -2.35720E-06
Element 26 of element group 1
Int point 1            -2.35720E-06
Element 27 of element group 1
Int point 1             7.07160E-06
Element 28 of element group 1
Int point 1            -2.35720E-06
Element 29 of element group 1
Int point 1             4.71476E-06
Element 30 of element group 1
Int point 1             4.71476E-06
```

```
Element 31 of element group 1

Int point 1            4.71476E-06

Element 32 of element group 1

Int point 1            4.71476E-06


*** End of list.
```

```
ADINA: AUI version 7.2.2, 14 May 1999:  *** NO HEADING DEFINED ***
Licensed from ADINA R&D, Inc.
Finite element program ADINA, response range type load-step:
Listing for zone WHOLE_MODEL:
Element field variables are evaluated using RST interpolation.
       POINT                  FORCE-R

Time 1.00000E+00

Element 1 of element group 1

Int point 1             3.01111E-17

Element 2 of element group 1

Int point 1            -3.01111E-17

Element 3 of element group 1

Int point 1             4.95050E-01

Element 4 of element group 1

Int point 1             4.95050E-01

Element 5 of element group 1

Int point 1             1.33408E-16

Element 6 of element group 1

Int point 1            -1.33408E-16

Element 7 of element group 1

Int point 1            -1.33408E-16

Element 8 of element group 1

Int point 1             1.33408E-16

Element 9 of element group 1

Int point 1             4.95050E-01

Element 10 of element group 1

Int point 1             4.95050E-01

Element 11 of element group 1

Int point 1            -1.33408E-16

Element 12 of element group 1

Int point 1             1.33408E-16

Element 13 of element group 1

Int point 1             1.48504E+00
```



```
Element 14 of element group 1
```

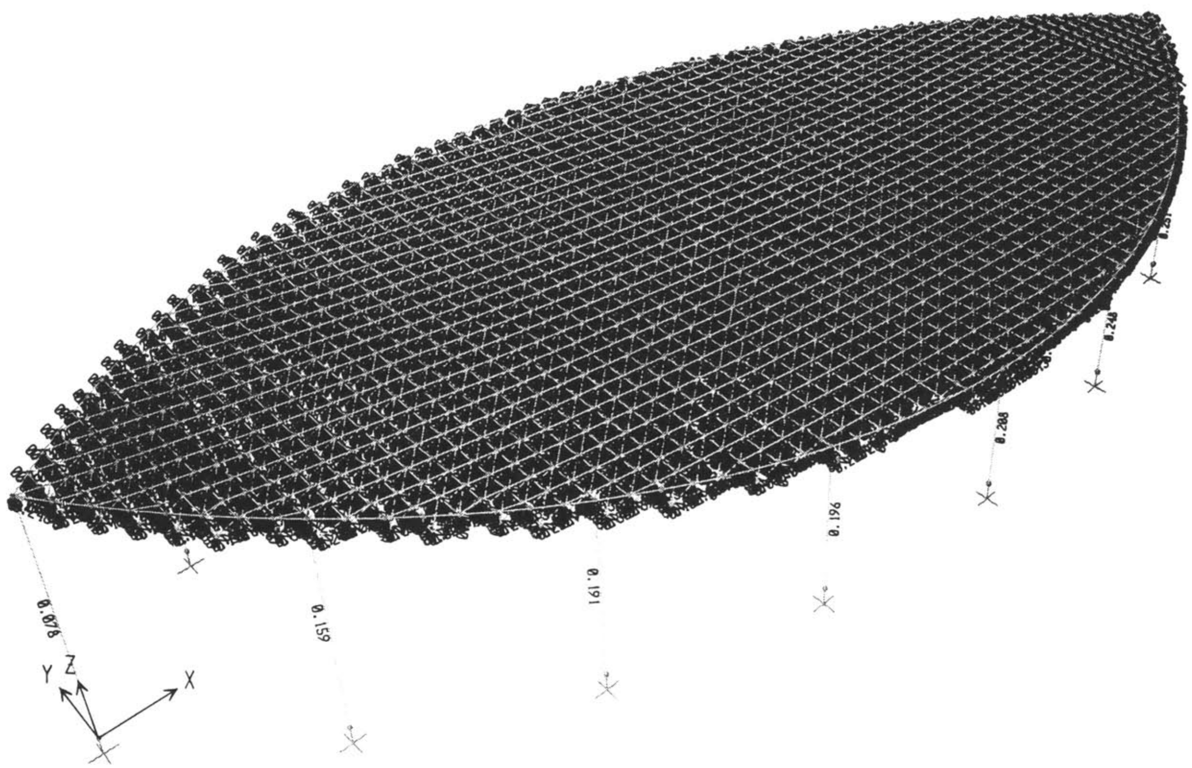
```
Int point 1              -4.95012E-01
Element 15 of element group 1
Int point 1              -4.95012E-01
Element 16 of element group 1
Int point 1              -4.95012E-01
Element 17 of element group 1
Int point 1              -4.95012E-01
Element 18 of element group 1
Int point 1               1.48504E+00
Element 19 of element group 1
Int point 1              -4.95012E-01
Element 20 of element group 1
Int point 1              -4.95012E-01
Element 21 of element group 1
Int point 1              -4.95012E-01
Element 22 of element group 1
Int point 1              -4.95012E-01
Element 23 of element group 1
Int point 1              -4.95012E-01
Element 24 of element group 1
Int point 1               1.48504E+00
Element 25 of element group 1
Int point 1              -4.95012E-01
Element 26 of element group 1
Int point 1              -4.95012E-01
Element 27 of element group 1
Int point 1               1.48504E+00
Element 28 of element group 1
Int point 1              -4.95012E-01
Element 29 of element group 1
Int point 1               9.90099E-01
Element 30 of element group 1
Int point 1               9.90099E-01
```

```
Element 31 of element group 1

Int point 1              9.90099E-01

Element 32 of element group 1

Int point 1              9.90099E-01


*** End of list.
```

APPENDIX C

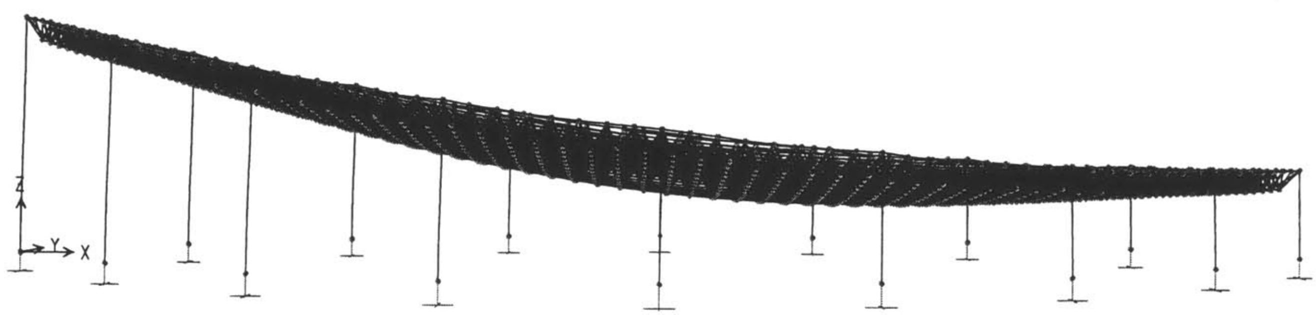
SAP2000

May 14,1999 12:50

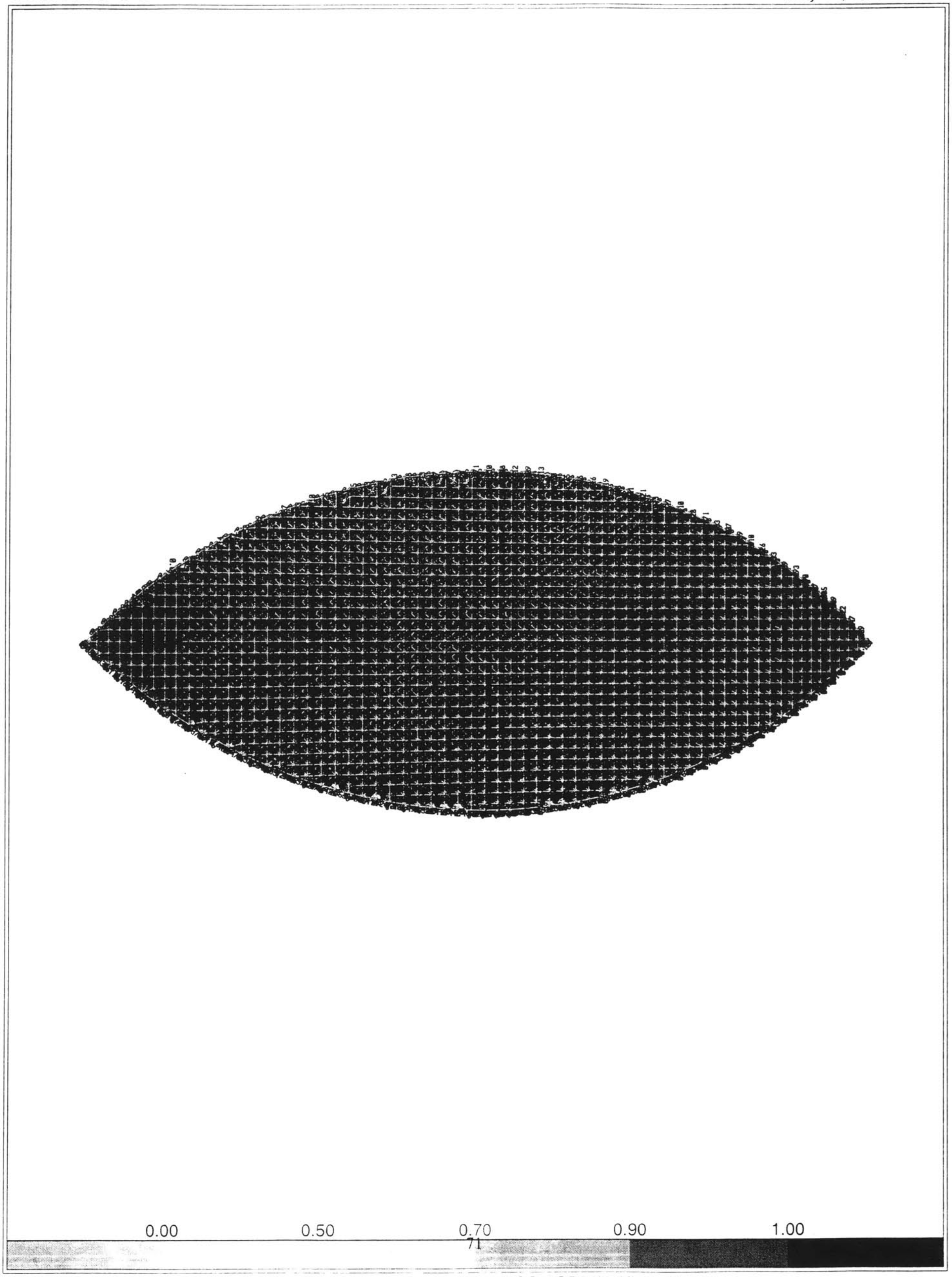


0.00 0.50 0.70 0.90 1.00

SAP2000 v6.13 - File:roof truss5 - Steel P-M Interaction Ratios (AISC-ASD89) - Kip-in Units



SAP2000 v6.13 - File:roof truss5 - Deformed Shape (DEAD) - Kip-in Units



SAP2000 v6.13 - File:roof truss5 - Steel P-M Interaction Ratios (AISC-ASD89) - Kip-in Units

REFERENCES

Nooshin, Hashyar. Formex Configuration Processing in Structural Engineer. Elsevier Applied Science Publishers, London and New York, 1984

Makowski, Z. S. Analysis, design and construction of double-layer grids. Applied Science Publishers Ltd, 1981

Nosshin, Hashyar. Formex Data Generation in International Symposium-Innovative Applications of Shells and Spatial Forms. Oxford & IBH Publishing Co, 1988

Nooshin, Hashyar. Basic Concepts of Formex Configuration Processing, International Symposium on Membrane Structures and Space Frames, Osaka, Japan, September 1986.

Disney, Peter. Formian: The programming Language of Formex Algebra in Shells, Membranes and Space Frames, Proceedings IASS Symposium, Osaka, 1986, Vol.3. Elsevier Science Publishers B.V., Amsterdam, 1986

Ik Nang Anna Cheah. Formices and Plenices in Structural Analysis and Design in Shells, Membranes and Space Frames, Proceedings IASS Symposium, Osaka, 1986, Vol 3. Elsevier Science Publishers B.V., Amsterdam, 1986.

Bathe, Dlaus-Jurgen. Finite Element Procedures Simon & Schuster / A Viacom Company, Upper Saddle River, New Jersey, 1996